

# BIM OG ENERGIANALYSE

OPTIMERING AF INGENIØRENS MODELLERINGSPROCESSER

Kandidatspeciale - Januar 2015  
cand.scient.techn - bygningsinformatik  
Aalborg Universitet

Peter Lind Johansen  
20130269

# TITELBLAD

---

**Titel**

BIM og Energianalyse - Optimering af Ingeniørens Modelleringsprocesser

**Forfatter**

Peter Lind Johansen

**Hovedvejleder**

Kjeld Svidt, Aalborg Universitet

**Bivejleder**

Rasmus Lund jensen

**Oplag**

Digital Publicering

**Udgivelsesdato**

08.01.2015

**Sprog**

Dansk

**Sidetæl**

106 (+ Appendiks, 29 s)

**Anslag**

148.148 (ekskl. Referencer, etc.)

<>

## RESUMÉ

Nærværende speciale er udarbejdet som afgangsprøve på Aalborg Universitets uddannelse for cand.scint.techn med speciale i bygningsinformatik.

Projektet har til formål at undersøge metoder til at optimere manuelle opmålingsaktiviteter forbundet energiberegning i Be10 som er påkrævet ved byggesagsbehandling i Danmark.

Undersøgelsen er gennemført med en naturvidenskabelig tilgang til teknologi-støttet procesoptimering på baggrund af et samarbejde med MOE A/S Rådgivende Ingeniører, som har stillet projektets primære case til rådighed.

Denne omhandler et udviklingsprojekt i virksomheden med fokus på udnyttelse af informationer fra BIM i forbindelse med Be10 beregning.

Projektprocessen omfatter en overordnet problemanalyse og herunder procesanalyser efter Contextual Design, som udgangspunkt for tests af metoder til interoperabilitet mellem Revit og Be10.

Problemanalysen indgår i valget af metode til en efterfølgende udvikling af et konkret teknologisk værktøj til af informationer fra BIM som grundlag for projektets konklusion.

## ABSTRACT

This study has been conducted as the final thesis at Aalborg University, MSc. Building Informatics

The project examines technological methods to optimize labor intense measurement activities associated with energy calculation in the energy analysis software Be10 as a requirement by Danish building regulations.

The study was conducted with a scientific approach to technology-facilitated process based on a collaboration with MOE A / S Consulting Engineers, which has provided the primary case for the study.

This deals with a company development project which is focused on the use of Building Model Information in Be10 calculations.

The project process includes a widespread problem analysis including process analysis based on the Contextual Design method, as a trigger to a research approach bases on tests of methods to achieve interoperability between Revit and Be10.

The problem analysis included in the choice of method for a subsequent development of a specific technological software application used for extracting information from BIM. The developement process and following tests acts as a basis for the project's final conclusion.

# FORORD

---

Projektet beskrevet i denne rapport er gennemført i perioden 01.09.2014 - 08.01.2015 og udgør afgangsprojetet på Aalborg Universitets uddannelse: cand.scient.techn med speciale i bygningsinformatik.

Projektet har til formål, at afdække udviklingsgrund for teknologi-støttet procesoptimering med fokus på ressource krævende opmålingsaktiviteter forbundet med Be10 beregninger.

I den forbindelse ønsker forfatteren at takke de bidragsydere, der har gjort projektet til en realitet:

Kjeld Svidt, Aalborg Universitet

- Hovedvejleder

Rasmus Lund Jensen, Aalborg Universitet

- Vejleder

Steffen Maagaard, MOE A/S Rådgivende Ingeniører

- Virksomhedskontakt

Jeremy Tammik, Autodesk Developer Network

- Teknisk vejledning

CUBO Arkitekter

- Testcase, virtuel bygningsmodel

## Begreber og betydning i teksten

### Add-in

Tilføjelse til software/program/plattform.

### API

Application Program Interface

### BEM

Building Energy Modeling

### BIM

Generel betegnelse for Bygnings Informations *Modellering* og *Modeller*, som teknologi, koncept, etc.

Betegnelsen kan være forbundet med forskellige kontekstbestemte betydninger:

- 1) Proces: Bygnings Informations *Modellering*.
- 2) Virtuel bygningsmodel: Bygnings Informations *Model*.
- 3) Verbum for relaterede processer, f.eks.: BIM-projektering.
- 4) Adjektiv for relaterede redskaber, f.eks.: BIM-værktøj.

I afsnit hvor begrebet har en specifik betydning, er det skildret eksplicit i teksten, som BIM-proces, BIM-model, BIM-objekt, etc.

### BIM-model

Bygnings Informations Model, som *virtuel bygningsmodel* og *information* der skabes, redigeres, udveksles og anvendes i forbindelse med *Bygnings Informations Modellering* som proces (BIM-proces). Omfatter modeller og informationer behandlet i værktøjer forbundet med BIM som teknologi. Herunder ligeledes modeller uden konkret bygningsgeometri, f.eks. uafgrænsede rum, landskab, etc.

### BIM-proces

Bygnings Informations Modellering, som *proces* med relaterede teknologiske værktøjer, samt eventuelt samarbejde.

### C#

CSharp - programmeringssprog

### CAD

Computer Aided Design.

### Class

Informationsstruktur/skabelon defineret for objektforekomster

### Compile/Kompilere

Computerhandling forbundet med oversættelse af programmeret kode til konkret program

### Data (modeling) language

Specifikt datasprog for opbevaring af informationer. F.eks. EXPRESS

### Datamodel

Repræsentation af data og struktur i en specifik struktur. F.eks. IFC eller gbXML

### EAM

Energy Analytical Model

### GUID

Globally Unique IDentifier. Unikt computergenereret ID

### Host

Objekt som et eller flere andre objekter eksisterer i. F.eks. Væg hoster vindue.

**IFA**

IFC File Analyzer

**Kildekode**

Original programmeringskode til et program

**Kodeblok**

Vilkårligt antal linjer af sammenhængende kode med et eller flere specifikke formål, i. Almindeligvis indrammet af 'braces':  
{ blok; }

**Modeldata**

Værdier og informationer relateret til objektattributter og egenskabsdata.

**Namespace**

Samling af specifikke classes i .NET

**Objekt (<Data-objekt/kode-objekt>)**

Objekter relateret til programmering jf. Objekt Orienteret Programmering forkortet OOP.

**Objekt (BIM-objekt)**

Byggeobjekt i virtuel bygningsmodel. Omfatter ethvert element med egenskaber, herunder inklusive rum eller områder i og omkring en virtuel bygningsmodel.

**OOTB**

Out Of The Box

**Performance**

Generelt begreb som dækker over ydelsesmæssige egenskaber forbundet med rum, bygningsdele, komponenter, etc.

**Plugin**

Se *Add-in*.

**Sample**

Prøve/eksempel

**Schema**

Specifikt defineret understruktur for specifikt datasprog. F.eks. IFC (EXPRESS schema)

**SDK**

Software Development Kit

**Udveksling**

Generel betegnelse for informationsudveksling.

**UI**

User Interface / BrugerGrænseflade i software.

**Workflow**

Arbejdsgange, procedurer og processer i sammenhæng med arbejdsstruktur, kommunikation, teknologi, etc.

## FIGURER

Figurer, diagrammer, modeller og tabeller uden kildehenvisning i figurteksten er udarbejdet af specialets forfatter.

# INDHOLD

<b>INTRODUKTION</b>	<b>1</b>
<b>1.1 INDLEDNING</b>	<b>2</b>
<b>1.2 TEORI OG METODE</b>	<b>4</b>
1.2.1 DATAINDSAMLING	4
<b>1.3 CASE 1</b>	<b>5</b>
1.3.1 MOE A/S	5
1.3.2 UDVIKLINGSPROJEKT	6
<b>1.4 INITIERENDE PROBLEM</b>	<b>7</b>
<b>PROBLEMANALYSE</b>	<b>11</b>
<b>2.1 PROCESANALYSE</b>	<b>12</b>
2.1.1 METODISK GRUNDLAG	12
2.1.2 ANALYSEBAGGRUND	15
2.1.3 WORKFLOW	17
2.1.4 DELKONKLUSION	24
<b>2.2 INTEROPERABILITET MELLEMLER CAD OG BE10: TEST AF METODER</b>	<b>25</b>
2.2.1 GRUNDLAG	25
2.2.2 CASE 2	29
2.2.3 TEST: GREEN BUILDING XML (GBXML)	32
2.2.4 TEST: INDUSTRY FOUNDATION CLASSES (IFC)	38
2.2.5 TEST: REVIT PLATFORM API	51
2.2.6 TEST: ROCKWOOL ENERGY DESIGN	65
2.2.7 DELKONKLUSION: INTEROPERABILITET OG TEST AF METODER	70
<b>2.3 PROBLEMFORMLERING</b>	<b>71</b>
<b>PROBLEMLØSNING</b>	<b>72</b>
<b>3.1 UDVIKLING AF EN REVIT ADD-IN</b>	<b>73</b>
3.1.1 OBJEKT ORIENTERET PROGRAMMERING (OOP)	73
3.1.2 MÅLSÆTNING	74
3.1.3 UDVIKLINGSTRIN	74
3.1.4 UDVIKLINGSMILJØ	75
3.1.5 PROGRAMMERING	76
<b>3.2 AFPRØVNING OG CASES</b>	<b>89</b>
3.2.1 ADD-IN OG FUNKTIONALITET	89
3.2.2 TESTMODEL CASE 2	90
3.2.3 KONTROL AF OUTPUT	91
3.2.4 CASE 3	94
<b>3.3 RESULTATER</b>	<b>95</b>
<b>3.4 DISKUSSION</b>	<b>96</b>
<b>KONKLUSION</b>	<b>97</b>





## 1.1 INDLEDNING

---

Energi, klima og miljø er med mere eller mindre en fast bestanddel af almindelige dagligdagsindtryk. Hvad enten det gælder sociale medier, internet, tv eller annoncering i offentlig transport, er global opvarmning genstand for opmærksomhed. Det presserende behov for at reducere CO<sub>2</sub>-udslip og forbruget af energiresourcer, er emner for samfundsmæssig såvel som politisk debat og ord som 'CO<sub>2</sub>-neutral' er almindeligt i beskrivelser, af alt fra websider til materialeudvikling og bygningsdesign. Der er stor fokus på miljø og energi i byggeriet, der er forbundet med adskillige betegnelser, som eksempelvis lavenergibyggeri, miljørigtigt byggeri, bæredygtigt byggeri, grønt byggeri og tilmed grøn BIM (Teknologisk Institut, 2014).

BIM (Building Information Modeling) som kollaborativ udviklingsproces og ressource til bygningsinformation har potentiale til at understøtte energioptimeret bygningsdesign. Det kan både være i forhold til menneskelig kommunikation og informationsudveksling mellem faglige aktører, og teknologisk kommunikation i form af dataudveksling mellem værktøjer, der støtter det menneskelige samarbejde. Et sandsynligt scenarie kan være analyse- og simuleringmodeller, der opererer på input fra bygningsmodeller. En forudsætning for dette er interoperabilitet og relaterer til flere aspekter. Herunder forenelighed af forskellige modellers konceptuelle betragtning af 'bygning' i sammenhæng de pågældende værktøjers behov for input. Det gælder eksempelvis for BIM som input i BEM (*Building Energy Modeling*). En given analyse med BEM kan have behov for specifikke input om uspecificerede forhold i BIM, som eksempelvis *Boundary Conditions*. Hvor BEM forholder sig til overflader, beskriver BIM typisk opbyggede konstruktioner, uden entydig definition af en analytisk grænse (Kim, et al., 2014).

Et andet aspekt af interoperabilitet er konkret dataoverførsel og værktøjers evne til at kommunikere med hinanden, som det grundlæggende element i forhold til anvendelighed.

Klima og energi i det politiske aspekt afspejler sig blandt andet i byggeriet, som regler for energiforbrug og myndighedskrav til dokumentation af energibehov. I Sverige er dokumentering af overholdte krav forbundet med konkrete målinger den fysiske opførte bygning. Det gør det nødvendigt for rådgivere, at sikre tilstrækkelig performance gennem analyse og simulering. Der er således ingen krav om dokumentation i forbindelse med designprocessen, hvilket overlader en vis frihed for rådgivernes valg af analyse- og simuleringværktøjer (Mondrup, et al., 2012).

De danske myndigheder anvender et anderledes koncept, hvor Bygningsreglementet/BR10 stiller krav til dokumentation af bygningers energibehov i forbindelse med byggeansøgning (Energistyrelsen, 2014). I modsætning til Sverige er det således et krav i designprocessen, og før byggeriet fysisk kan begynde. BR10 stiller krav om energidokumentation i henhold til *DS418 Beregning af bygningers varmetab* (Dansk Standard, 2011) og *SBI-anvisning 213 Bygningers energibehov*, der beskriver en beregningsvejledning med anvendelse af programmet *Be10* (Aggerholm & Grau, 2011).

En række input i en Be10-beregning er baseret på opmålinger af bygningsgeometri (Aggerholm & Grau, 2011). I BIM-CAD (*Computer Aided Design*) er en virtuel bygning aggregeret af BIM-objekter, der repræsenterer fysiske bygningsdele, elementer og komponenter (Eastman, et al., 2011). BIM-modeller kan derfor potentielt udgøre en omfangsrig informationskilde til input i analysemodeller og optimering af manuelle aktiviteter. Potentialet forudsætter, under fællesbetegnelsen *interoperabilitet*, at softwareprogrammer, BIM- og analysemodeller er i stand til at 'tale et fælles sprog'. Herunder, at BIM-informationer kan rumme, og omsættes til, input i overensstemmelse med analytiske betragtninger af bygninger, og ikke mindst at informationer rent teknisk kan overføres imellem forskellige værktøjer. Simulering og analyse med værktøjer som eksempelvis *IDA ICE* (EQUA, 2014) og *IESVE* (IES, 2014) kan være nødvendige i forbindelse med rådgiveransvaret for et svensk byggeri. Disse værktøjer betragter og beregner en bygning forskelligt fra Be10 (Mondrup, et al., 2012). De kan derfor i princippet både producere virkelighedstro resultater og være BIM-kompatible, men samtidigt overflødige i forhold til den Be10-beregning, som i sidste ende er gyldig dokumentation. Under disse omstændigheder kan det i stedet være relevant, at fokusere på BIM og interoperabilitet i sammenhæng med Be10, og automatisering af input, som teknisk middel til procesoptimering. I første omgang for energi- og indeklimatekniøeren til en hurtigere og mindre ressourcekrævende analyseproces, og i sidste ende til gavn for hele projektorganisationen som grundlag for energioptimeringer, tidligere i designprocessen.

## 1.2 TEORI OG METODE

---

Interoperabilitet i forhold til input i Be10 og BIM som primær ressource, kan som udgangspunkt være forbundet med to overordnede indgangsvinkler. En samfundsvidenskabelig og en naturvidenskabelig. Nærværende undersøgelse beskæftiger sig med den naturvidenskabelige. Derfor er den metodiske tilgang primært baseret på konkrete forsøg, med bagvedliggende litteraturstudier. Min problembetragtning er imidlertid inspireret af (Leavitt & Bahrami, 1988) med den overbevisning, at et problem skal betragtes i en kontekst. På denne baggrund har jeg anvendt *flow models* (Beyer & Holtzblatt, 1998) i en kontekstanalyse. I den forbindelse har en tidlig udgave indgået i en dialog med case-virksomheden, som grundlag for en revision og kortlægning af projektets kontekst. Den reviderede flow model indgår i udformningen, af en efterfølgende forsøgsbaseret problemanalyse, som omfatter en række metoder til interoperabilitet og ligger til grund for en problemformulering og problemløsningsfase. Problemløsningsfasen fokuserer på konkret udvikling af et funktionelt løsningskoncept til informationsudtræk fra linkede BIM-modeller.

### 1.2.1 DATAINDSAMLING

Sekundære eksterne data er indhentet fra litteratur, primært i form af videnskabelige artikler og websider. Formålet med denne kombination er et informationsgrundlag med forskningsbaseret pålidelighed, suppleret med ny information om teknologisk udvikling og produktspecifikke data. Dette omfatter ligeledes regulativer i form af lovgivning og standarder.

Sekundære interne data fra den primære case har jeg indsamlet i udleveret materiale i form af præsentationsmateriale og kompendier for virksomhedsprocedurer. Samtidig har jeg indsamlet primære kvalitative data i forbindelse med projektmøder og dialog med en virksomhedsrepræsentant.

Undersøgelsen er gennemført som casestudier af en primær virksomhedsbaseret case, suppleret med en sekundær case i form af en virtuel bygningsmodel. Den primære case er styrende for kontekst, procesanalyse, fokusretning og vurdering af relevans, og danner således grundlag for modelleringen af den virtuelle bygning, der udgør den sekundære case. En betydelig del af de analyserede data beskrevet i rapporten, stammer fra mine forsøg med denne virtuelle bygningsmodel.

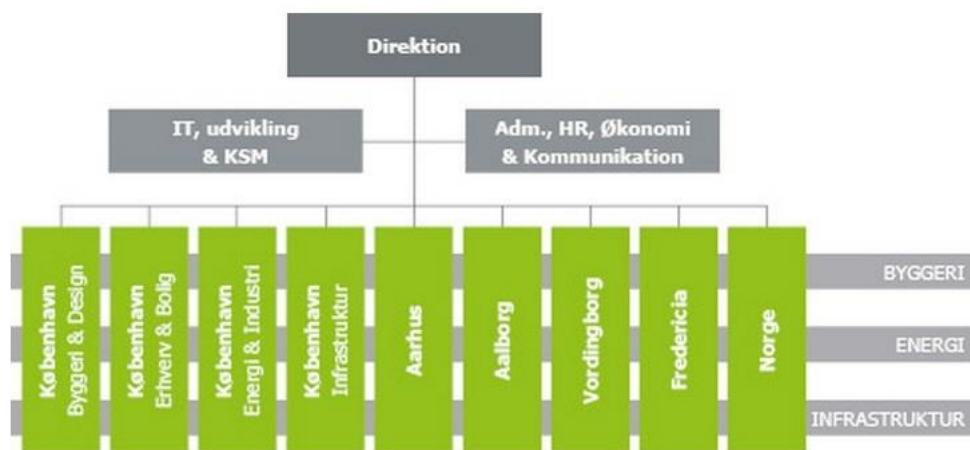
## 1.3 CASE 1

Afsnittet introducerer undersøgelsens primære case, der omhandler et udviklingsprojekt i Aarhus-afdelingen af MOE A/S Rådgivende Ingeniører.

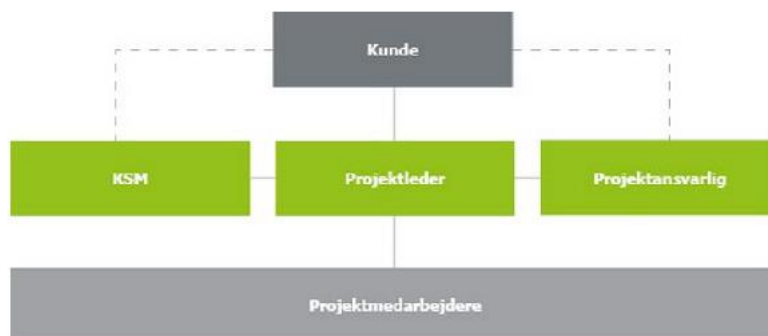
### 1.3.1 MOE A/S

#### ORGANISATION

MOE er en nordisk koncern med 5 afdelinger i Danmark og et i Norge. Virksomheden beskæftiger mere end 500 ansatte indenfor 3 forretningsområder, som fordeler sig på byggeri, energi og infrastruktur (MOE, 2014a).



Figur 1.3.1. MOE Basisorganisation (MOE, 2014b)



Figur 1.3.2. MOE Projektorganisation (MOE, 2014b)

#### VIRKSOMHEDSUDVIKLING

Byggeri udgør det største forretningsområde, hvori virksomhedens udvikling fokuserer på nye rådgiverydelser, udbygning af lønsomhed i projekter og reduktion af projektomkostninger (MOE, 2014a).

Som et led i dette indledte virksomhedens Aarhus-afdeling i 2013 et udviklingsprojekt med henblik på optimering af Be10 beregning med udgangspunkt i BIM.

### 1.3.2 UDVIKLINGSPROJEKT

Virksomheden arbejder løbende med kompetenceudvikling og udnyttelse IKT til omkostningsreduktion. I den forbindelse virksomhedens udviklingsprojekt rettet mod informationsudtræk fra bygningsmodeller, som til reducere af ressourceforbruget til Be10-beregninger. Disse har traditionelt været forbundet med manuelle opmålinger af projekt materialet, som både kan være tidskrævende og uoverskueligt i store projekter. Behov og vision for virksomhedsprojektet, er i det følgende opsummeret som rammer og mål, i vilkårlig rækkefølge:

1. Udgangspunkt i Revit og linket arkitektmodel.
2. Reduktion af tid og omkostninger forbundet med opmåling, som samtidigt kan være uoverskueligt for store byggerier.
3. Automatisk opmåling af overfladearealer, vinduer og orienteringer baseret på udtræk af viden/information der eksisterer i BIM.
4. Samspil med ventilationsprojektering.
5. Sammenhæng med skemastruktur i Be10.
6. Opmåling i overensstemmelse med DS418.
7. Sempelt template anvendelig for medarbejdere med større fokus på kernekompetence end teknologi.
8. Automatisk data-eksport til Excel.
9. Udviklingsgrundlag for rutiner til at automatisere udveksling med Be10.

Derudover er hurtigere tilbagemelding på konsekvenser af designændringer og udtræk til indeklima- og dagslysberegning beskrevet som sekundære mål (MOE, 2014c).

Virksomhedens efterfølgende udviklingsarbejde har medført en procedure, der overordnet består af ingeniør/arkitekt koordineret parameteropsætning i Revit med eksterne plug-ins, en Revit-template med opsætning og skema-eksport til Excel med tabeller linket til Be10. Revit-templaten består af foruddefinerede skemaer for:

1. Bruttoetageareal
2. Vægge
  - 2.1. Ydervægge
  - 2.2. Facadepaneler
3. Tag
4. Gulve
5. Fundamenter
6. Vinduer
7. Yderdøre
8. Skygger
9. Ventilation
10. Belysning

Den indeholder ligeledes opsætning til zone- og arealplaner, som kan oprettes til modellering af spaces med opdeling i belastningszoner og bruttoetagearealer.

Det modellerede etageareal opmåler samtidig en omkreds, som giver information om fundamentlængde i stueplan. Proceduren nedbringer tidsforbruget til opmåling betydeligt, men er afhængig af indledende dialog med arkitekten, tæt koordinering med BIM Manageren og supplerende beregninger i Excel (MOE, 2014c).

## 1.4 INITIERENDE PROBLEM

---

Afsnittet redegør for min egen problematiserende beskrivelse af udviklingsprojektet i casen, ud fra en faglig betragtning af tekniske elementer og proces. Denne problematisering leder op til en formulering af et initierende arbejdsproblem.

En øget informationsmængde og variation hænger sammen med at BIM-værktøjer er tilgængelige for stort set alle faggrupper. Virtuelle BIM-modeller repræsenterer fysiske bygninger, som en samling af byggeobjekter, der repræsenterer fysiske bygningsdele. De virtuelle bygningsdele kan opbevare en række associerede egenskaber eller attributter, der kan beskrive alt fra visuel fremtoning og geometrisk udformning til energimæssig performance. Når arbejdsformen bliver mere udbredt involverer det også faglige roller med ansvar for specifikke dele af den samlede informationsmængde. Men hvordan udnytter man denne ressource, når informationer med forskellige faglige relationer, er bundet op på en enkelt BIM-objekttype?

Denne problematik kan være forbundet med en række potentielle løsningsmuligheder med forskellig fokus, herunder: 1) Informationsbehov. En aktør med ejerskab for et givent objekt, udfører supplerende ydelser, tilføjer og leverer informationer i overensstemmelse med andre aktørers behov. 2) Automatisering ved hjælp af teknologiske tiltag/værktøjer. 3) Ekstern kobling. Tilknytning af eksterne informations-/datasæt efter et 'klæbe-koncept', hvor supplerende informationer kan forbindes med objekter, for eksempel via krydsreferencer til GUID, eller lignende. 4) Fælles objekt-database. BIM-objekter med delte egenskabsdatasæt, hvor forskellige aktører kan tilknytte fagspecifikke informationer, som de hver især har ansvar for.

Informationsbehov er omfattet af IDM-standarder for procesplanlægning og informationsleverancer (buildingSMART, 2014). Men i forbindelse med Be10, kan det være svært at argumentere for at den virtuelle bygningsgeometri er utilstrækkelig i forhold til målbare parametre eksempelvis vinduesorientering, idet informationer er repræsenteret i geometrien, men ikke nødvendigvis som læsbare talværdier. Fokus på informationsbehov er her betragtet som tværfaglig udnyttelse af information, faciliteret af menneskelig handling og kommunikation, herunder ydelsesansvar og aftaleforhold. De øvrige er i højere grad rettet mod tekniske faktorer, der omfatter funktionalitet i software på hver sin måde. Automatisering er betragtet som supplerende funktionalitet, der kan udtrække tilgængelige data fra modtaget materiale i en given situation, uden behov for at forudgående tilknytning af særlige informationer. Ekstern kobling er betragtet som en kobling mellem eksterne modeller og et lokalt supplement af informationer, som er i stand til at håndtere nye udgaver af eksterne modeller. Fælles objekt-database er betragtet som eksempelvis BIM-objekter med dele egenskabsæt, hvor forskellige aktører kan tilknytte fagspecifikke informationer, som de hver især har ansvar for. Sidstnævnte kan favorisere dynamiske samarbejdsformer, som eksempelvis cloud-løsninger, hvor informationer tilføjes og opdateres real-time. Dette kan udfordre procedurer for procesdokumentation og de mere statiske projektweb-løsninger som, i skrivende stund, er den lovmæssigt foreskrevne norm for IKT i offentligt byggeri, særligt §5 i IKT-bekendtgørelsen (Klima-, Energi- og Bygningsministeriet, 2013).

## OBJEKTBASERET PARAMETRISK MODELLERING

Projektering med i Revit og BIM-værktøjer generelt er betragtet efter betegnelsen *Objektbaseret Parametrisk Modellering* i henhold til (Eastman, et al., 2011):



*The technology on which most BIM design applications are based. Includes the ability to define individual objects whose shape and other properties that can be controlled parametrically.*

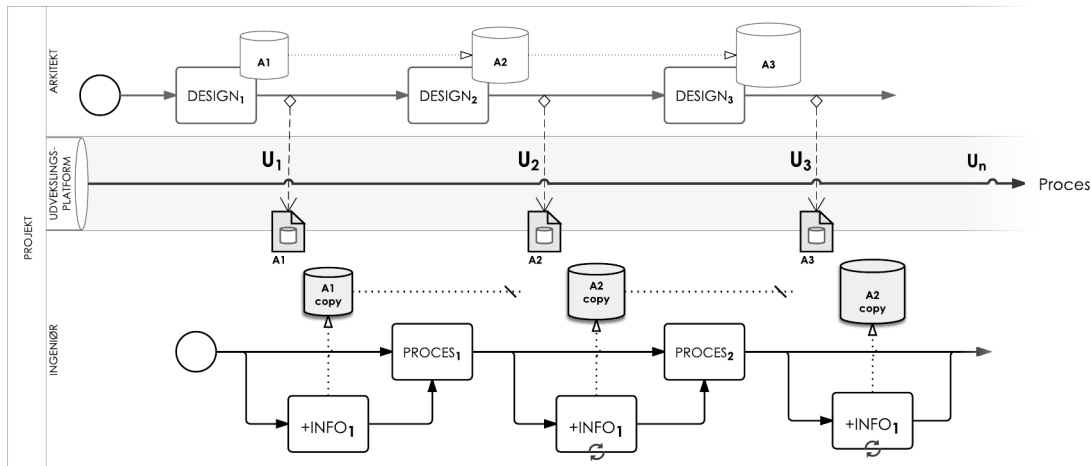
*(Eastman, et al., 2011, p. 589)*

Bygningsdele som døre og vinduer indgår som en ikke ubetydelig del af en bygnings overordnede udtryk, form og funktion. I et teknisk perspektiv, tilhører de repræsentative bygningsdele/BIM-objekter, den specifikke model de er modelleret/placeret i. Typisk en arkitektmodel for døre og vinduer. I den forstand er en arkitekt ansvarlig for at eksempelvis form og størrelse er projekteret i overensstemmelse med gældende regulativer. Døre og vinduer spiller imidlertid en rolle for langt mere end arkitektonisk udtryk. I en konceptuel betragtning, er de gennembrydninger af bygningens klimaskærm. Det gør sig blandt andet gældende for energi- og indeklimate, hvor informationer om blandt andet glas- og u-værdier, er et ingeniørfagligt ansvarsområde.

Den nuværende samarbejdsform er baseret på udvekslingsplatforme som projektwebs efter forskrifter i IKT-bekendtgørelsen (Klima-, Energi- og Bygningsministeriet, 2013). Den udvekslingsform skaber en række barrierer for udnyttelse af information baseret på BIM-objekter. En sådan barriere er uoverensstemmelser mellem grænseflader for faglige ansvarsområder og ejerskab af data. Idet vinduesobjekter er placeret i arkitektmodellen er ejerskabet, i et teknisk perspektiv, placeret hos en given arkitekt, selvom ansvar for egenskaber er fordelt mellem flere aktører i virkelighedens verden. I en BIM- over tid, er vinduer og associerede informationer, kun tilgængelige i en skrivebeskyttet 'læse-tilstand' for andre end den pågældende ejer i det tekniske perspektiv. Forstået på den måde, at fagmodeller, hver især udgør et originalmateriale for et specifikt fagområde, som er '*den gældende*' information i den samlede projektkontekst. Informationsudveksling mellem faglige aktører, via platforme som projektwebs, relaterer til dokument-koncept for udveksling af data. Forstået på den måde at relevante aktører, hver især kopier egne BIM-fagmodeller og uploader som statiske dokumenter til det fælles projektarkiv. Hver kopi repræsenterer i princippet et statisk billede, afskåret fra bagvedliggende proces og information. Konceptet kan betragtes som en *dokumentbaseret tilgang* til informationsudveksling. Dette kombineret med skarpt optrukne grænser for ejerskab af data, kan gøre det problematisk at opnå konsensus i forhold til objektorienterede BIM-processer. En dokumentbaseret tilgang til udveksling af BIM-data forhindrer ikke en enkelt aktør i at redigere en modtagen model, som eksempelvis en ingeniør der tilføjer glasværdier til vinduesobjekter. Det betyder derimod, at pågældende supplement, kun er forbundet med et projektmateriale i det øjebliksbillede, som en given modeludveksling repræsenterer. Selvom en aktør isoleret set, er fagligt ansvarlig for den specifikke information, er pågældende aktør ude af stand til, at tilføje den i et overordnet projekt, hvis ejerskabet, i et objektorienteret perspektiv, tilhører en anden. Med andre et ord, kan en efterfølgende modeludveksling siges at slette supplerende informationer, idet en eksisterende model erstattes med en ny.



Et sådant scenarie er relateret til flere problematikker. For det første er en vis sandsynlighed for utilsigtede ændringer umulig at udelukke. Det kan potentielt føre til fejlbehæftet projektmateriale. For det andet kan det være præget af en høj grad af redundans, hvor et givent supplement udelukkende er gyldigt indtil næstkommende modeludveksling, hvorefter det må tilføjes igen. Hvis projektets parter eksempelvis udveksler en gang hver uge, bliver en og samme aktivitet således en ugentlig begivenhed. Dette er illustreret i figuren:



Figur 1.4.1. Konceptuel procesdiagram: Dokumentbaseret udveksling af modeller i en principiel designproces for arkitekt og ingeniør

Elementer i den opstillede model angiver følgende:

- A** Arkitektmodel i designprocessen.
- A<sub>copy</sub>** Kopi af arkitektmodel.
- U<sub>n</sub>** Tidspunkt for informationsudveksling.

Rektangulære kasser angiver aktiviteter af vilkårlig varighed:

- DESIGN<sub>n</sub>** Arkitektproces der leder op til en given udveksling.
- PROCES<sub>n</sub>** Ingeniørproces med udgangspunkt i modtagen arkitektmodel.
- +INFO** Aktivitet forbundet med at tilføje supplerende information i en modelkopi.

Optrukne linjer angiver procesforløb.

Stiplede linjer angiver information overført gennem udvekslingsplatform.

Prikkede linjer symboliserer lokalt informationsflow for en aktør.

**+INFO** fremgår i begge tilfælde som **1** og angiver en gentagelse af aktiviteten. Informationer tilføjet i den første modelkopi, kan ikke føres tilbage til det overordnede projekt og må tilføjes igen. Ethvert supplement kan dermed siges at strandes ved en efterfølgende udveksling.

Linkfunktionen i Revit danner et teknisk grundlag for at opbevare informationer skabt ud fra en modtaget model og genbruge disse i forbindelse med opdaterede modeller. Forudsætningen for dette er imidlertid, at informationer tilføjes i host-filen. For det hidtil beskrevne koncept, kan det eksempelvis omfatte zones og spaces, som er afgrænset af geometrien i det linkede arkitektprojekt.

Denne form for workflow forebygger utilsigtede ændringer af modtaget information, da softwaren forhindrer en bruger i at redigere i links. Det ændrer ikke på at den principielle proces er præget af redundans.

På den ene side afhjælper funktionen kun problematikken omkring grænseflader for ansvar og overlappende objekttilknytning af informationer, i begrænset omfang.

På den anden side er det uvist, om det nogensinde har været en del af udviklernes målsætning. Muligheden for at linke modeller fra forskellige fagdiscipliner, danner grundlag for den kollaborative BIM-designproces i Revit. Den er stærkt medvirkende til at samarbejdende byggefaglige aktører, er i stand til at anvende hinandens producerede information på tværs af en projektorganisation. At denne informationskilde i det hele taget er tilgængelig, er samtidigt årsagen til at de pågældende faggrupper er i stand til at identificere informationer, som er *utilgængelige*. Dette paradoks er et eksempel på, at nye problemer kan rejse sig fra løsningen af eksisterende. I den forstand er kollaborative BIM-processer oplagte genstande for undersøgelser i forhold til '*Aalborg-modellen for problembaseret læring*' (Aalborg Universitet, 2011).

Mulige løsninger på denne problematik kan eksempelvis omfatte cloud-baserede samarbejdsformer [se eksempelvis (Eastman, et al., 2011) om *BIM-servere/-repositories*].

I forhold til udviklingsprojektet i virksomhedscasen samt CAD-værktøjer og IKT-bekendtgørelser taget i betragtning, kan sådanne løsningsmuligheder være relevante i et fremtidsperspektiv, hvorimod interoperabilitet og automatiseret dataudtræk i højere grad fokuserer på resultater i et mere kortsigtet perspektiv.

Dette argument ligger til grund for arbejdsspørgsmålet formuleret i det følgende.

## INITIERENDE PROBLEMFORMULERING

### **1. *Hvordan skabes en kobling mellem Revit og Be10, som kan automatisere manuelle opmålinger og samtidigt reducere behovet for supplerende information i lokale modelkopier?***

Begrebet *kobling* dækker i denne sammenhæng over teknisk interoperabilitet såvel som arbejdsprocesser i relation til problemet.

Kapitlet beskriver projektets problemanalyse i de overordnede dele: Procesanalyse og Interoperabilitet mellem CAD og Be10 - og test af metoder  
De to afsnit leder op til projektets problemformulering.



**ALT VIDEN LIGGER I MODELLEN -  
VI SKAL BARE TRÆKKE DEN UD**

*-INGENIØR, ENERGI- OG INDEKLIMA*

## 2.1 PROCESANALYSE

Citatet på kapitelforsiden relaterer både til måden de enkelte informationer er repræsenteret i en BIM-model og udvikling af informationer gennem specifikke input fra rådgiverfaglige delprocesser i relation til ansvarsgrænseflader og redundans, som beskrevet tidligere. Afsnittet indledes med en introduktion af min metodiske tilgang til den efterfølgende procesanalyse. Denne følges af den grundlæggende baggrund for teknologiske og processuelle aspekter i designprocessen. Denne er analyseret i principielt perspektiv, med fokus på faglige aktører, aktanter og roller i et workflow forbundet med at skabe BIM-modeller med nødvendige informationer til en Be10 beregning. Det principielle perspektiv på processen har til formål at rette en efterfølgende opmærksomhed på typisk eksisterende emner, som med stor sandsynlighed kan forekomme i fremtidige projekter og således er relevante at optimere så videst muligt.

### 2.1.1 METODISK GRUNDLAG

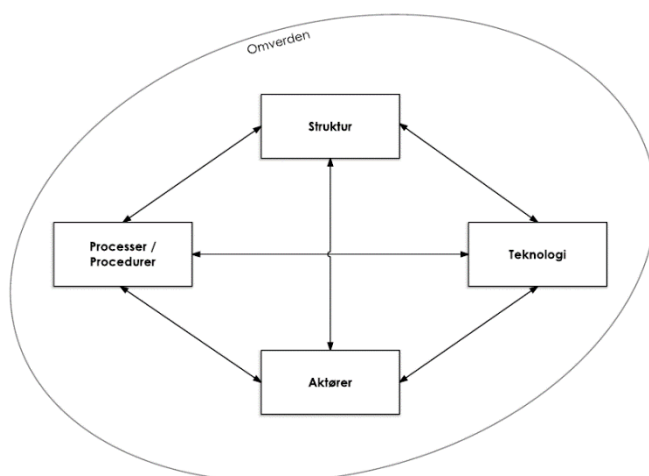
Det metodiske grundlag for procesanalysen er herunder beskrevet i to dele. Den første del omhandler metodernes teoretiske beskrivelser, hvorefter deres anvendelse i procesanalysen er beskrevet.

#### 2.1.1.1 LEAVITT'S ORGANISATIONSMODEL

Leavitt's model, oprindeligt formuleret i 1965, beskriver en organisation som bestående af 4 tæt forbundne hovedelementer: Struktur, mennesker, opgaver og teknologi (1988). Ifølge Hildebrandt (2011) er modellen primært en forandringsmodel, hvor elementerne er relaterede, så ændringer af ét element vil påvirke de øvrige. De 4 hovedelementer i den oprindelige model, er siden suppleret med et kulturbegreb og omgivelser (Hildebrandt, 2011). Elementerne kan beskrives således:

- Struktur: Arbejdsfordeling, organisering og kommunikationsveje i virksomhed/afdeling
- Aktører/mennesker: Medarbejdere, faglighed og personlige værdier
- Opgaver: Virksomhedens kerneforretning eller specifikke funktioner
- Teknologi: Hjælpemidler til at udføre opgaver. Kan principielt omfatte alt fra bygninger til maskiner eller informationsteknologi.

Modellens hovedpointe er sammenspillet mellem elementerne (Bejder & Olsen, 2007).



Figur 2.1.1. Leavitt model [efter (Leavitt & Bahrami, 1988)]

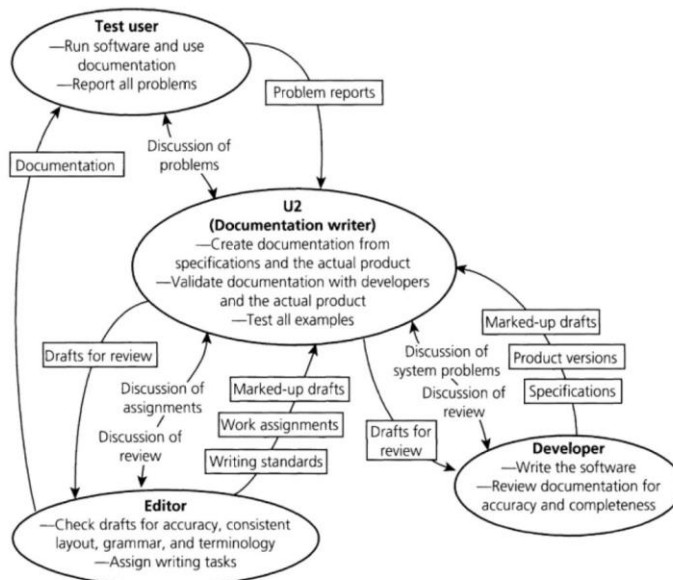
### 2.1.1.2 CONTEXTUAL DESIGN

Contextual Design henvender sig som udgangspunkt til systemdesign, hvor funktionalitet er et produkt af en brugercentreret designproces. Det grundlæggende koncept beskriver indsigt i brugeres eksisterende systemer, som en forudsætning for at designe et funktionelt system. Et funktionelt system kan optimere arbejdsprocesser ved at understøtte og strømline brugerhandlinger. Metoden behandler en arbejdsproces efter en systembetragtning. Et system kan således lige så vel være en sammensætning af menneskers manuelle handlinger og aktiviteter, som det kan være menneskers interaktion med software. Metoden lægger vægt på at kortlægge og beskrive arbejdsprocesser i et forståeligt sprog, som kan indgå i en dialog mellem designer og bruger. Dette sprog kalder forfatterne A Language of Work og består af grafiske opstillinger i en række konceptuelle modeller. (Beyer & Holtzblatt, 1998)

En flow model fokuserer på at afdække workflow i en arbejdssituation. Modellen kan beskrive følgende:

- Individ: Personer eller roller, der udfører arbejdet. Vises i oval cirkel
- Individens ansvar: Oplistet i oval cirkel
- Grupper: Markering af personer med fælles mål eller handlinger.
- Flow: Kommunikation mellem personer/roller. Samtale, koordinering eller udveksling af artefakt. Vises med pile.
- Artefakter: Fysiske eller virtuelle forekomster i arbejdet. Vises i kasser på flow-pile.
- Emne: Kommunikation uden udveksling af artefakt. Skrives til flow uden kasse
- Steder personer går til eller fra i forbindelse med arbejdet: Vises i stor kasse med stedets navn.
- Nedbrud eller problem i kommunikation eller koordinering: Vises med et lyn.

(Beyer & Holtzblatt, 1998, p. 91)



Figur 2.1.2. Eksempel på flowmodel (Beyer & Holtzblatt, 1998, p. 93)

## METODER I PROCESANALYSEN

Leavitt (1988) lægger vægt på, at alle organismer i et system principielt påvirker hinanden. Samtidigt kan systemet udsættes for ydre påvirkninger fra omverdenen. Modellen kan eksempelvis anvendes til at kategorisere elementer i en projektorganisation. For mit vedkommende er den mest af alt en kognitiv påvirkning, der argumenterer for at ethvert projekt må betragtes i dets kontekst. For at være i stand til dette, har jeg anvendt Contextual Design (Beyer & Holtzblatt, 1998), mere specifikt *Flow Models*, som værktøj til at analysere og visualisere projektets kontekst i forhold til arbejdsgange, roller og aktører/aktanter.

Min fortolkning af elementer i flowmodellen adskiller sig, i begrænset omfang, fra Contextual Design, hvor artefakter er beskrevet som noget personer udveksler i deres kommunikation og *places* er *fysiske* steder personer kan bevæge sig hen. Jeg har valgt at inddrage en mellemting af disse, som et supplement. Dette kan være virtuelle rammer, hvor informationer opbevares og bearbejdes, manuelt eller automatisk. Med andre ord, har jeg inddraget teknologiske elementer som aktanter i flowmodellerne. Herunder CAD-værktøjer, analyse- og simuleringsværktøjer, servere etc.

## 2.1.2 ANALYSEBAGGRUND

Revit som et teknologisk aspekt i procesanalysen, der bygger på data indsamlet gennem præsentationsmateriale, virksomhedsdialog og en række udleverede kompendier, der beskriver relevante virksomhedsprocedurer og anvendelse af teknologi. Foruden Revit, som beskrevet i det efterfølgende afsnit, omfatter baggrundsmaterialet (MOE, 2014c) og referater i Appendiks A og B. Udleverede kompendier er holdt fortrolige af hensyn til virksomhedens interne forhold.

### 2.1.2.1 REVIT

Revit er parametrisk i den forstand, at objekter er styret af parametre i forskelligt omfang. Det gør det muligt at tilpasse objekter ved at ændre parameterværdier. Parametre kan være brugerdefinerede eller indbyggede. Nogle indbyggede parametre er synlige i brugergrænsefladen, hvor en bruger manuelt kan aflæse eller tilskrive værdier. Andre er indbyggede som systemparametre, der blandt andet definerer relationer, der gør objekter i stand til at tilpasse sig hinanden. Det kan eksempelvis være åbninger i vægge, hvor vinduer er placeret. Systemparametre håndteres automatisk i takt med brugerhandlinger og modellering. (Autodesk Inc., 2014a). Brugerdefinerede parametre kan anvendes til supplerende informationer som tekst eller specifik type, som længde og areal med flere (Autodesk Inc., 2014g).

#### FAMILIES

Alle objekter eller elementer i Revit-sammenhæng, er generelt betegnet som *Families*. Betegnelsen er forbundet med begreber som blandt andre *Category*, *Type* og *Instance*.

#### System Families

Primære bygningsdele. Herunder vægge, gulve og tage. System Families er integreret i softwarens objektstruktur, og kan interagere, eksempelvis ved samlinger mellem vægge og gulve.

#### Modeling Families

Geometriske objekter, der repræsenterer fysiske bygningsdele eller komponenter, uafhængigt af en specifik model. Herunder vinduer og døre. Modeling Families indeholder basisdefinitioner af grundlæggende geometri, parametre og egenskaber. For eksempel *fastkarmsvindue med en lodret og to vandrette sprosser fordelt på 1/3 af vindueshøjden*.

#### Category

Overordnet bygningsdelstype. Definerer hvordan et BIM-objekt indgår i den virtuelle bygningskontekst. Det kan eksempelvis være en betragtning af at bygningsdele i kategorien *vinduer*, som udgangspunkt fungerer som åbninger placeret i primære bygningsdele, der afgrænser bygningens rumgeometri.

#### Family Type

Variant af en given Family. Eksempelvis ovenstående med en bredde på 900mm og en højde på 1200mm. Families kan indeholde flere foruddefinerede typer med forskellige dimensioner og egenskabsværdier.

**Type Properties:** Egenskabsværdier defineret i en Family Type, som gælder for enhver forekomst af den pågældende type, placeret i en model.

### **Family Instance**

Én specifik forekomst af en given Family Type.

**Instance Properties:** Værdier relateret til den specifikke forekomst. Eksempelvis etage og brystningshøjde.

(Autodesk Inc., 2014b)

Princippet kan med udgangspunkt i et vindue opsummeres til følgende:

- Family er den overordnede skabelon og definerer grundlæggende udformning og hvordan et vindue kan påvirkes af input.
- Family Type er en udfyldt skabelon med værdier for de tilgængelige inputmuligheder.
- Family Instance et specifikt vindue efter den udfyldte skabelon, som er placeret i modellen.

### **LINKEDE MODELLER**

Revit's link-funktion kan sammensætte delmodeller i en overordnet model, eller tilføje modtagne fagmodeller fra samarbejdende aktører. Softwaren gør det muligt at læse informationer i linkede modeller og udføre kollisionskontroller, men forhindrer brugere i at foretage konkrete ændringer (Autodesk Inc., 2014f).



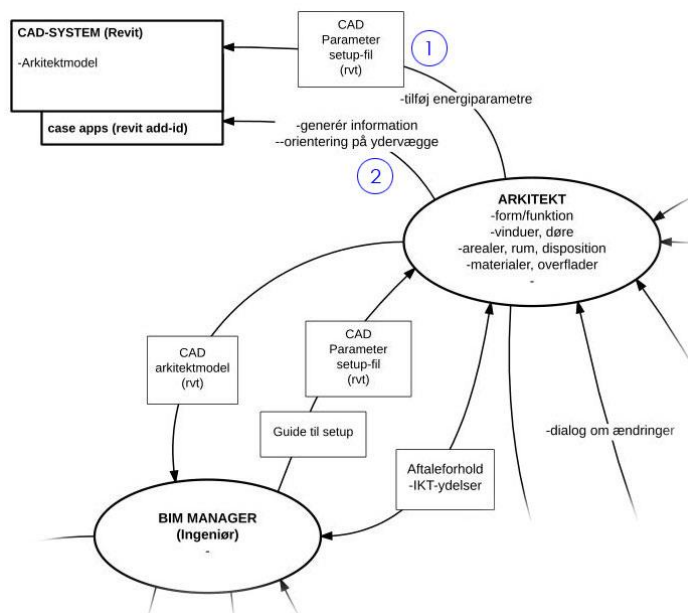
### 2.1.3 WORKFLOW

I det følgende beskrives den analyserede proces med udgangspunkt i energi- og indeklimaingeniøren og relationer til Be10-beregning. Aktiviteter, procedurer og handlinger er beskrevet med supplerende udklip fra flowmodeller processen er analyseret ved hjælp af. Analysens fokus er særligt rettet mod teknologiens rolle i processen og herunder de aktiviteter og menneskelige handlinger, der er forbundet med anvendelse af teknologi. Både i form af teknologi som fremmer processen, men i særdeleshed aktiviteter og handlinger forbundet med at gøre teknologi anvendelig.

Processen er analyseret med energi- og indeklimaingeniøren i centrum, men omfatter mere end relationer og informationsudvekslinger. Nogle medarbejdere har supplerende roller som *CAD-operatører*, der sørger for koordinering og opsætning af modtagne fagmodeller. I flowmodellen er aktiviteter forbundet med denne rolle, er beskrevet under en overordnet *BIM Manager*-rolle i workflowet. Flowmodeller udarbejdet i forbindelse med analysen findes i fuld udgave i Appendiks C.

Tilføjelsen af de nødvendige informationer om vinduesorienteringer er forbundet med en kompleks procedure, der involverer flere aktører og roller. Hver især udfører en række delaktiviteter, der ender ud i parametre med informationer om orienteringer og hældninger. Disse aktiviteter er beskrevet selvstændigt i det følgende.

Indledningsvis er parterne nødt til at indgå aftaler om supplerende IKT-ydelser for arkitekten. Når aftaleforhold er på plads og designprocessen er i gang, udleverer ingeniørens BIM-manager en Revit-fil til arkitekten. Filen er suppleret med et kompendie med vejledning til at opsætte energimæssige parametre ved hjælp af udleverede fil.

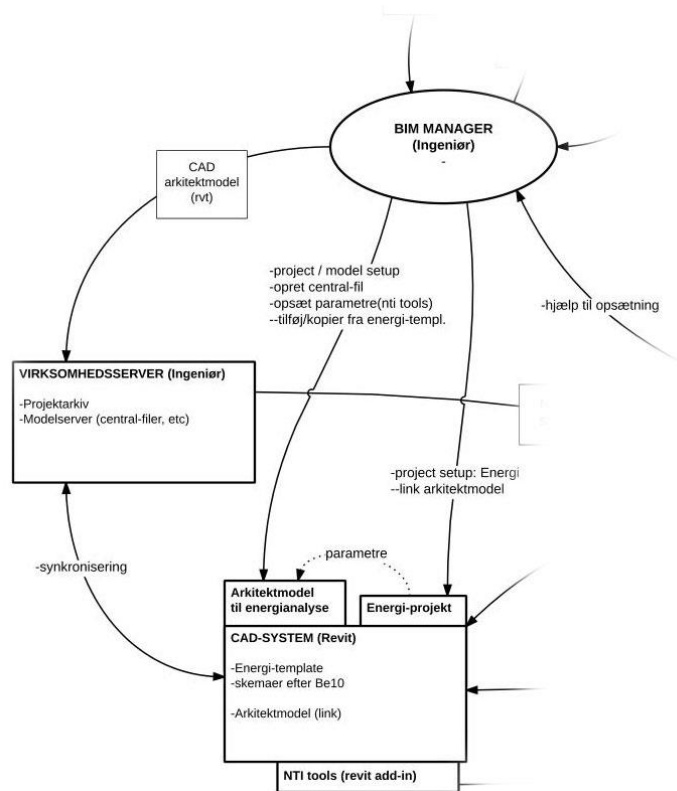


Figur 2.1.3.1. Workflow: Supplerende arkitektydelse for (1) energiparametre og (2) vægorientering.

Opsætningen tilføjer 12 parametre, fordelt på henholdsvis vægge, vinduer, døre og facadepaneler. Hver type tilføjes *Orient Angle*, *Orient Text* og *ID\_from\_host* efterfulgt af typeindikator, eksempelvis *Orient Angle (Y)* for ydervægge. Efter opsætningen af parametre tilføjer arkitekten orienteringer i de nyoprettede parametre for ydervægge. Kompendiet vejleder i den forbindelse om, hvordan

aktiviteten kan udføres med et gratis plug-in, ved navn CASE Apps (CASE Design, 2014) Dette plug-in registrerer modellens ydervægge, hvorefter arkitekten manuelt skal aktivere en '*record direction*'-funktion, som skriver den registrerede retning til en given væg. Aktiviteten udføres to gange. Én for vinkel (f.eks. 90°) og én som tekst (f.eks. N). Arkitektmodellen er dermed klargjort til ingeniørerne.

I forbindelse med en efterfølgende udveksling, gemmer ingeniørens BIM Manager arkitektmodellen som central model.

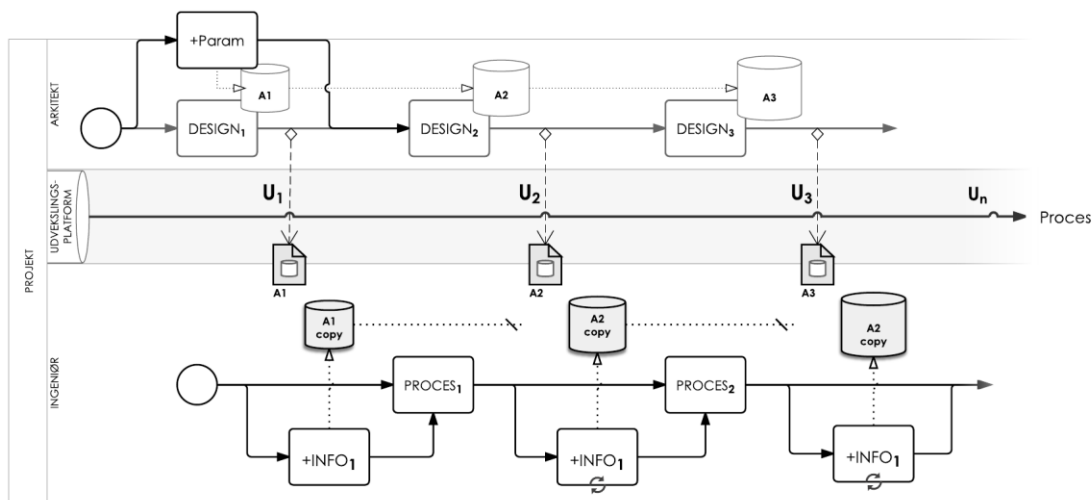


Figur 2.1.3.2. Workflow: Håndtering og anvendelse af modtagne arkitektmodeller.

Den lokale kopi bearbejdes med yderligere opsætning af parametre, inden den linkes til et energiprojekt. Denne opsætning udføres med *NTI Tools* (NTI, 2014), som kan kopiere parametre og værdier fra en given ydervæg til hostede vinduer og døre. På den måde er ingeniøren i stand til at udfylde de energi-parametre arkitekten har oprettet for vinduer/døre/facadepaneler, med de orienteringer arkitekten har tilføjet til ydervægge. Disse parametre er tekstbaserede og har derfor ingen indflydelse på arkitektprojektet, som sådan. Dermed undgår ingeniøren, at foretage egentlige ændringer og sikrer samtidig, at data omkring beskriver bygningens geometri, stammer fra arkitekten.

For arkitektens vedkommende er proceduren principielt en engangsydelse, medmindre en eller flere vægge og/eller vinduer/døre bliver erstattet. For ingeniørerne er proceduren derimod forbundet med hver modeludveksling, som ifølge kompendiebeskrivelser typisk finder sted en gang ugentligt (pr. projekt).

Dermed har den modtagne arkitektmodel, en række sammenfald med konceptet for lokale arbejdskopier i den principielle proces beskrevet i afsnit 1.4. Dette er illustreret i nedenstående figur:



Figur 2.1.3.3. Konceptuelt procesdiagram: Principielt designproces med fokus på redundans i den analyserede kontekst.

**+Param** Arkitektprocedure for parameteropsætning og tilknytning af vægorientering.

Arkitektproceduren for opsætning af parametre og tilknytning af vægorienteringer, er symboliseret af **+Param**. Når først opsætning er gennemført og informationer er tilføjet af arkitekten, er de som udgangspunkt tilstede i projektet fremadrettet. I så fald er de tilgængelige for de beskrevne ingeniøraktiviteter, som derimod gentager sig. Disse er repræsenteret af **+INFO**. På trods af en relativt omfattende proces for én specifik parameter, kan det i nogle projekter være nødvendigt for ingeniøren, at skildre vinduers individuelle lokalisering for at vurdere at skyggeforhold. Virksomheden har i forbindelse med en *BIM AARHUS* konference i februar 2014, præsenteret et konkret eksempel på en sådan problematik, i form af vinduer i gårdrum (MOE, 2014c). I det beskrevne eksempel, er vinduer identificeret ved at modellere midlertidige rum (*Rooms*) ved facaders udvendige side og aflæse rumrelationer i egenskaber for de enkelte vinduer. En sådan *work around*, må ligeledes foretages i lokale kopier, idet data tilskrives vinduesobjekter - selvom de genereres af softwaren.

## ENERGI OG INDEKLIMA

I forbindelse med den første udveksling i processen opretter ingeniør/BIM Manager et Revit-projekt efter en særlig energi-template. Templateen indeholder en række foruddefinerede skemaer, der er udformet efter tabeller i Be10. Indledningsvis indeholder energi-projektet ingen konkret model, men opstiller parametre og værdier fra arkitektmodellen, der tilføjes som link og danner grundlag for supplerende BIM-objekter. Begrebet energi-model er derfor anvendt i det følgende.

Termisk indeklima, energi og dagslys er indbyrdes relaterede og bearbejdes derfor, så vidt muligt, sideløbende. Foruden energiberegning og Be10, er termisk indeklima baseret på simuleringer og dagslys er baseret på beregning. Konstruktive opbygninger og U-værdier bygningsdele bestemmes i dialog med arkitekt og konstruktionsingeniør.

De energi-parametre, der er tilknyttet i den modtagne arkitektmodel indgår i energimodellens foruddefinerede skemaer. Energi- og indeklimaingeniøren er i stand til at eksportere disse skema til Excel-regneark med en funktion i NTI Tools. Det er i den forbindelse nødvendigt at bearbejde de eksporterede data manuelt i Excel, inden de kan anvendes som input Be10. Det kan eksempelvis omfatte sammenlægning af arealer eller lignende. Dette forudsætter samtidig en manuel tilpasning af celleformateringer, idet Revit eksporterer skemaer som tekst (Autodesk Inc., 2014c).

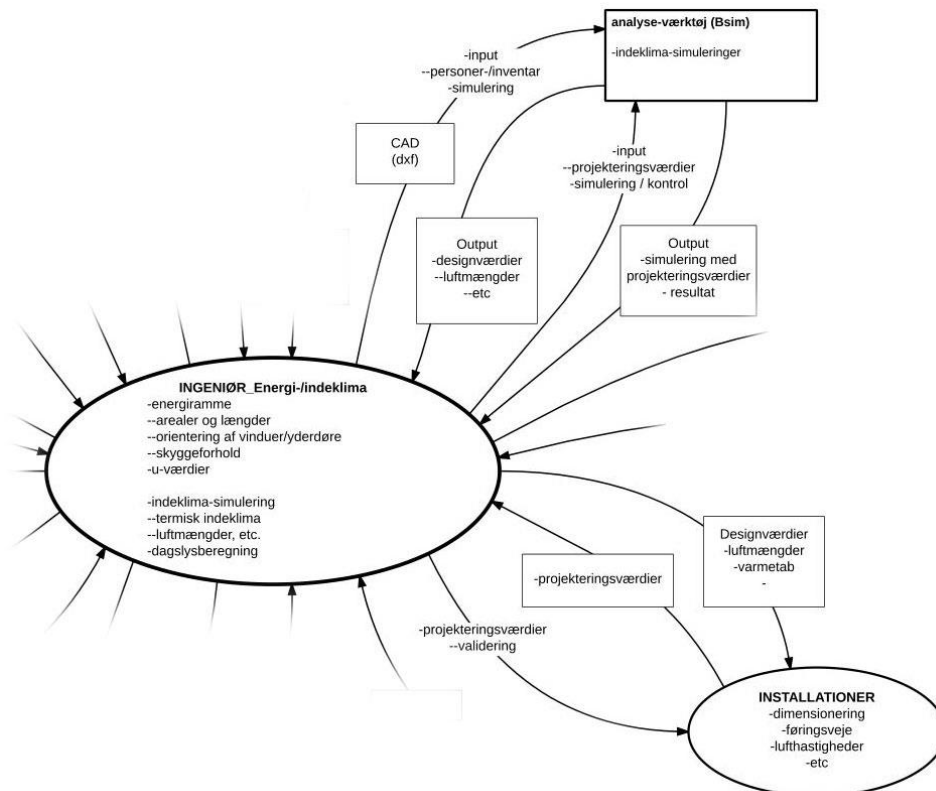
Dette gør sig ligeledes gældende for Excel-eksport med NTI Tools<sup>1</sup>. Excel er linket til Be10 med en API-løsning udviklet af virksomheden selv.

Ingeniøren tilføjer arealplaner og modellerer arealer til bruttoopmåling. Den linkede model indstilles som *'Room Bounding'* og kan dermed danne rammer for rummelige objekttyper, som *spaces*. *Spaces* udgør de analytiske områder i en bygning og tilføjes i energimodellen med den linkede arkitektmodel som afgrænsende geometri. *Spaces* differentierer sig fra *Rooms* gennem tilknyttede performance-data for eksempelvis luftskifte og relaterer i højere grad til funktion end indretning. Ét *space* kan således omfatte flere indretningsmæssige rum med samme funktion. Derfor er *spaces* som oftest forskellige fra de enkelte rum.

Performancedata for eksempelvis luftskifte, stammer fra simuleringer af bygningens termiske indeklima, med input for bl.a. person- og inventarbelastninger, fra rumbeskrivelser, byggeprogram, etc. Simuleringen foretages af energi- og indeklimaingeniøren, med værktøjet BSim (SBI, 2013). De enkelte værdier er i den forbindelse klassificeret som designværdier og danner grundlag for ventilationsprojektering. Det resulterer efterfølgende i projekteringsværdier, som kan være forskellige fra designværdier. De indgår derfor i en ny indeklima-simulering, hvor de kan valideres som tilstrækkelige eller give anledning til justeringer.

---

<sup>1</sup> Observeret for en testfil eksporteret med NTI Tools v. 2015.1



Figur 2.1.3.4. Workflow: Indeklima-simulering, ventilationsprojektering, design- og projekteringsværdier

Ingeniøren samler spaces i zoner med ensartet lokalisering og performance. Typedefinition, kombination og zoneinddeling er funktionsbestemt og baseret på faglige vurderinger.

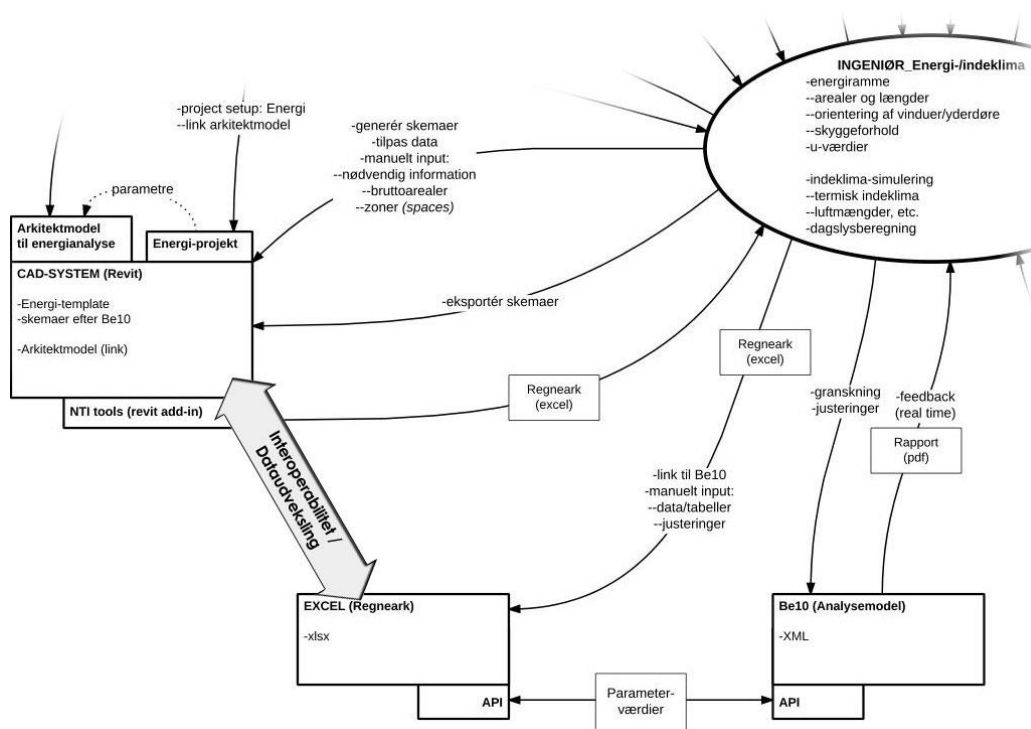
En sådan vurdering omfatter ligeledes en fortolkning af begreber i arkitektmodellen, hvor flere rum, hver især være navngivet forskelligt, men repræsenterer den samme funktionelle type i forhold til indeklimate og energibehov.

Selvom energimodellen gør brug af geometri i arkitektmodellen, er objekter og informationer ingeniøren tilføjer ikke præget af redundans i samme forstand som informationer i den lokale kopi.

I forbindelse med efterfølgende modeludvekslinger, kan en opdateringsprocedure typisk forløbe på følgende måde for BIM Manageren:

1. Downloader model fra projektweb
2. Erstatte eksisterende model med downloadet model
3. Gemmer som central model og udfører parameter-opsætning
4. Genindlæser link i energi-model.

Når den linkede model genindlæses, er softwaren i stand til at håndtere den downloadede version, som en opdatering og dermed bevares informationer tilknyttet energimodellen. Spaces kan således tilpasse sig eventuelle geometriske ændringer. Modellerede arealer kan imidlertid være nødvendige, at tilrette manuelt.



Figur 2.1.3.5. Workflow: Excel/Be10-link, performedata og manuelle input i BIM

Perfomedata skabt som resultat af energi-analyse, indeklimate-simulering og dagslys-beregning, kan være relevante at udveksle i BIM-modeller, eksempelvis som egenskabsdata for *spaces*.

Herunder design- og projekteringsværdier som udveksles med ventilationsingeniøren. Disse kan være forbundet med: 1) Manuelt input i Revit. 2) Redundans svarende til proceduren for parameteropsætning. Sidstnævnte gør sig kun gældende for informationer, som tilføjes i lokale kopier. Det kan eksempelvis være glasværdier for vinduer.

Fokus i den initierende problemformulering, relaterer sig til interoperabilitet og dataoverførsel, som illustreret af den dobbeltrettede pil i figuren ovenfor og omfatter primært *output fra Revit som input til Excel*.

#### 2.1.4 DELKONKLUSION

Processen bærer præg af redundans, som forekommer i forskellige afskygninger. Den primære årsag er det analysemæssige behov for orienteringsinformationer. Teknisk bunder det i lokale modelkopier og opsætningsprocedurer. Disse kan relateres til den parameterbaserede tilgang til skemaer, data og værdier, som implicit er påvirket af de benyttede it-værktøjer. Dette afspejler sig i en række konsekvenser, som mere eller mindre er direkte afledt af kopi-problematikken og kan medføre en 'overflod' af parametre og kopierede informationer.

Processen er imidlertid også forbundet med problematikker og redundans, som ikke er direkte afledt af konsekvenserne for lokale kopier. Det omfatter manuelle aktiviteter som justering af data og celleformatering, input og udveksling af beregnings-, analyse- og simuleringsresultater,

De supplerende arkitekttydelser kan give anledning til debat om honorarer i forbindelse med parternes aftaleforhold og ydelsen udgør reelt kun en mindre del af parameteropsætningen. En del som i princippet kan udføres som en del af ingeniørens opsætning, uden at medføre konsekvenser af særlig betydning. Den konkrete fordeling af aktiviteter til opsætning, er nødvendig for at holde proceduren generisk, men i sidste ende kan den reelle værdi begrænse sig til at arkitekten figurerer som kilde. Redundans kan potentielt nedbringes af projektspecifikke procedurer, hvis arkitekten ligeledes abonnerer på NTI Tools og dermed er i stand til at gennemføre opsætning i faktiske arkitektmodel. I forhold til figur 2.5, kan en sådan fordeling af ydelser stort set eliminere '+INFO'-elementer i ingeniørprocessen, men er sandsynligvis forbundet med en øget koordineringsindsats.

Virksomheden er opmærksom på de redundante aktiviteter, men proceduren har samlet set reduceret tidsforbruget, som er mere end halveret i forhold til hidtidige opmålingsprocesser. Dette retfærdiggør gentagelserne i proceduren. Manuelle aktiviteter er usandsynlige at eliminere fuldstændigt og aktiviteter som opdatering af arealer, medvirker til en faglig granskning, der i forvejen er forbundet med rådgiverfagligt ansvar.

I stedet for at fokusere optimering af processen i den nuværende afskygning, kan det være relevant at benytte en anden indgangsvinkel målrettet specifikke værdier frem for skemaer og parametre. Denne tilgang fokuserer på de informationer, som er repræsenteret i BIM-modellers geometri, samt tilgængelighed af disse informationer i form af automatisk udtræk eller eksport - uafhængigt af supplerende ydelser, ansvar og aftaleforhold.



## 2.2 INTEROPERABILITET MELLEML CAD OG BE10: TEST AF METODER

Citatet i indledningen af dette kapitel indrammer essensen af dette afsnit med få ord, om end i et forenklet perspektiv. Udtræk af informationer fra BIM er forudsat interoperabilitet mellem CAD- og analyseværktøjer, og eventuelle melleml. Interoperabilitet er forbundet med flere metoder med forskellig tilgang til informationsudveksling.

Indledningsvis er 4 metoder til interoperabilitet grundlæggende beskrevet, som udgangspunkt for efterfølgende tests og analyse. Dette er suppleret med en gennemgang af Be10 -parametre og relevans i forhold til informationstræk fra BIM, som efterfølgende leder op til en introduktion af projektets sekundære case, inden test og analyse af de enkelte metoder beskrives.

### 2.2.1 GRUNDLAG

#### 2.2.1.1 METODER TIL INTEROPERABILITET

For at teste interoperabilitet mellem Revit og Be10 har jeg gennemført individuelle forsøg med de 4 konkrete metoder, der tilsammen er betragtet som repræsentative i forhold til egnede og tilgængelige muligheder på tidspunktet for undersøgelsen. Disse omfatter følgende:

- Green Building XML (gbXML)
- Industry Foundation Classes (IFC)
- Revit Platform Application Program Interface (API)
- Rockwool Energy Design

IFC og gbXML er udvekslingsformater, der kan overføre informationer mellem software ved hjælp af eksport og import (Eastman, et al., 2011). IFC er international standard for udveksling af BIM data og gbXML har med udbredt implementering i software opnået status som de facto standard for overførsel af BIM-data til energianalyseværktøjer (buildingSMART, 2013a) (Stumpf, et al., 2011). Begge er *schema*-baserede formater, hvilket principielt betyder, at de hver især definerer et specifikt begrebsdomæne som håndteres gennem et overordnet datasprog. Det kan eksempelvis være i forbindelse med eksport, hvor den indbyggede datastruktur i et CAD-værktøj, oversættes til XML (*eXtensible Markup Language*) for gbXML og EXPRESS eller XML for henholdsvis IFC og IFCXML. Et *schema* fortæller softwaren hvorledes data skal skrives - og læses i det pågældende format. I hvert tilfælde sker en oversættelse af data ved eksport til disse formater.

Interoperabilitet gennem API kendetegner en direkte kommunikation med/mellem software og er deraf betegnet *Direct Link* (Eastman, et al., 2011). Kim, et al. beskriver en nødvendighed for manuel tilpasning af BIM-data som grundlag i energi-analyse og foreslår en API løsning til at specificere forhold for eksempelvis *Space Boundaries*, der udgør en analytisk defineret afgrænsning af rum og eller *spaces*, som kan variere i placering i forhold til tykkelse i en geometrisk afgrænsning (2014).

Rockwool Energy Design adskiller sig fra de øvrige, som et selvstændigt værktøj til energiberegning i overensstemmelse med Be10 (ROCKWOOL, 2013). Værktøjet understøtter informationsudtræk gennem integration i Revit (Dalux, 2012).

### 2.2.1.2 FREMGANGSMÅDE

Be10-parametre, der i første omgang er relevante, identificeret og opstillet i en prioriteret liste. Af hensyn til projektets varighed, indgår den prioriterede liste som observationsgenstand, hvoraf en specifik parameter er udvalgt som analysens drivkraft og primære fokus.

Hver metode er testet under de samme forudsætninger ud fra en testmodel som undersøgelsens sekundære case. Kvalifikationer er bedømt efter en række opstillede kriterier, beskrevet i det følgende.

#### KRITERIER

Kriterier til bedømmelse af metoder:

- Pålidelighed: Leverer gyldige data/resultater
- Værdi: Optimere proces/ressourceforbrug/økonomi
- Tilgængelighed: Databehandling i Excel
- Procesunderstøttelse: Kompatibel med Be10-relevant BIM-data
- Fremtidssikring: Fremadrettet udvikling og kompatibilitet med Revit versioner
- Implementering: Automatisere aktiviteter med færrest mulige step
- Tovejskommunikation: Potentiale for import af Excel/Be10-data.

Pålidelighed og værdi er altafgørende. En kvalificeret metode skal levere korrekt output og optimere tids- eller ressourceforbrug i forhold til eksisterende procedurer. Andet er irrelevant. Den skal samtidig være i stand til at overføre BIM-informationer, som er relevante for Be10.

Implementering og tovejskommunikation er betydende faktorer for optimeringspotentiale frem for egentlige kriterier. En metode skal så vidt muligt, lade medarbejdere fokusere på kernekompetencer og forebygge redundans.

## KORTLÆGNING AF RELEVANTE BE10-PARAMETRE

De relevante parametre er bestemt i samarbejde med virksomheden og bygger på repræsentantens faglige indsigt, konkrete erfaringer og faglige vurdering af potentielt værdiskabende dataudtræk fra BIM-modeller. Udeladte parametre kan eksempelvis være forbundet med dialogbaserede beslutninger eller lignende. Forhold omkring de relevante parametre er hver især genstand for opmærksomhed, selvom forsøg og analyse er gennemført med fokus på en specifik parameter i output.

Dertil hører også faktorer, som er forbundet med Be10 i indirekte forstand. Herunder identificering af bygningsdeles lokaliteter til vurdering af skyggeforhold, Be10-input til BIM og optimeringspotentiale i et fremtidsperspektiv. Sidstnævnte retter sig overordnet mod dagslys- og skyggeberegninger. Parametrene er beskrevet i samlet omfang i Appendiks D.

Et udsnit af parametrene er opstillet i de efterfølgende tabeller efter nedenstående princip:

[Overordnet Be10 kategori]			
Uddata	Inddata	Be10 parameter	
	x	[Parameter]	[Type]
x		<i>[Parameter]</i>	<i>[Type]</i>

Tabel 2.2.1. Beskrivelsesprincip for prioriterede Be10-parametre. Type repræsenterer parameterens primære association til BIM: Geometri (G) eller Egenskab (E)

Felter i tabellen repræsenterer følgende:

- Uddata: Input i BIM - fra Be10  
Inddata: Input i Be10 - fra BIM
- Prioritet: Talværdi i to niveauer med 1 som højeste og 2 som næsthøjeste.
- Type: Parameterens primære association i BIM  
G = geometri  
E = egenskab

*Kursiv skrift markerer en relevant parameter i fremadrettet perspektiv.*

## BE10-FOKUSPARAMETRE

Tabel 2.2.2 beskriver et udsnit med højest prioriterede af de kortlagte Be10-parametre.

Bygning			
U	I	Parameter	
		2 Opvarmet etageareal	G
		2 Opvarmet kælder	G
Ydervægge, tage og gulve			
		Parameter	
		2 Transmissionsareal	G
		Transmissionskoefficient (U)	E
Fundamenter og samlinger ved vinduer			
		Parameter	
		2 Kuldebro længde	G
		2 Linjetab	G
Vinduer og yderdøre			
U	I	Parameter	
		1 Antal	G
		1 Orientering*	G
		1 Hældning	G
		1 Transmissionsareal	G
		Transmissionskoefficient (U)	E
		Glasandel	G
		Rudens solvarmetransmittans	E
		Skygger	G
		Solafskærmning	G
Internt varmetilskud			
		Parameter	
		2 Areal	G
		2 Personer [W/m <sup>2</sup> ]	E
		2 Apparat i brugstiden[W/m <sup>2</sup> ]	E
		2 Apparat udenfor brugstiden[W/m <sup>2</sup> ]	E

Tabel 2.2.2. Relevante Be10-parametre for BIM-dataudtræk (repræsentativt udvalg) og markerede fokusparametre.

### ARGUMENTATION FOR PARAMETERFOKUS

Den redundans procesanalysen beskriver i forbindelse med vinduesorienteringer taler sit eget sprog. Den ene parameter, flere dele projektorganisationen i et eller andet omfang. Hvis ikke det er opmålingsaktiviteter, er det informationsleverancer og aftaleforhold. Informationer om orienteringer er visuelt repræsenteret i CAD-værktøjer, hvilket beviser at data eksisterer. De supplerende ydelser og parameteropsætninger kan dermed sidestilles med at kopiere information fra ét sted til et andet, i én og samme model, og efterfølgende kopiere, den kopierede information igen. Den ene parameters relative betydning i en samlet projektkontekst, er i sig selv et argument for at analysere optimeringsmuligheder. Hovedargumenterne for at fokusere på netop denne parameter kan opsummeres til følgende:

1. Sandsynlighed for at opnå konkrete resultater idet informationernes eksistens er bevist.
2. Potentiel basis for automatisering med udtræk af tilgængelige data.
3. Relation til flere projektaspekter. Både før og under designprocessen, i form af kontrahering og opmålingsaktiviteter.

## 2.2.2 CASE 2

De fire metoder til interoperabilitet er testet i individuelle forsøg med en bygningsmodel, som er modelleret med det specifikke formål, at indgå i undersøgelsen som sekundær case. Hensigten med dette er at være i stand til at udelukke uforudseelige påvirkninger i forbindelse med designudvikling og besidde en forhåndsindsigt, der kan bidrage til en mere målrettet og dybdegående, teknisk analyse. Forudsætningerne for de gennemførte forsøg med den sekundære case er beskrevet i det følgende afsnit, hvorefter fakta om det virtuelle bygningsmodel er opsummeret.

### 2.2.2.1 TESTMILJØ

De fire forsøg er gennemført med et grundlæggende setup bestående af følgende:

#### OPERATIVSYSTEM

- 64-bit, Windows 7 Professional, Service Pack 1

#### BIM PLATFORM<sup>2</sup>

- Revit 2015, Update Release 4

#### BIM APPLIKATIONER

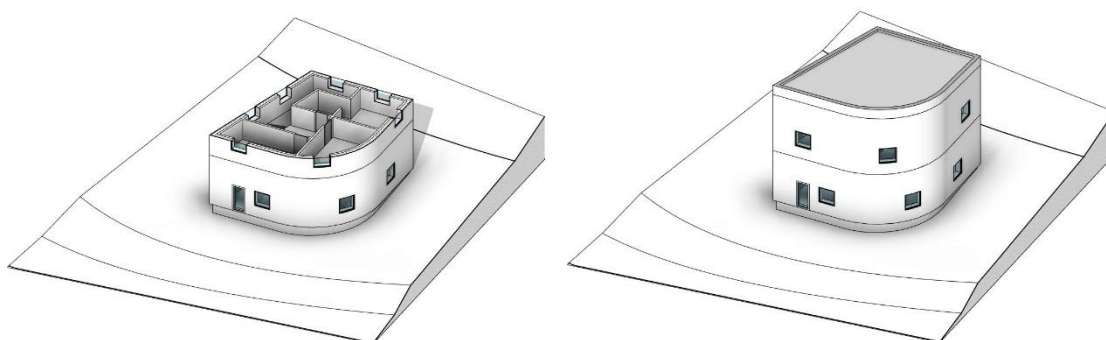
- DDS-CAD Viewer (Data Design System, 2014): Visuel gennemgang af geometri og modelldata.
- IFC File Analyzer v.2.24 (NIST, 2014): Kategoriseret Excel-beskrivelse af *entities* i IFC-model.

#### ANDEN SOFTWARE

- Notepad++ (Ho, 2014): Gennemgang af gbXML- og IFC-datamodel
- Microsoft Visual Studio 2012
- Microsoft Office/Excel 2013

### 2.2.2.2 VIRTUEL BYGNINGSMODEL

Den virtuelle bygning er relativt simpel og af begrænset størrelsesmæssigt omfang. For at inddrage en vis kompleksitet omfatter facadeforløbet dog andre vinkler end 90° og en rundet del.



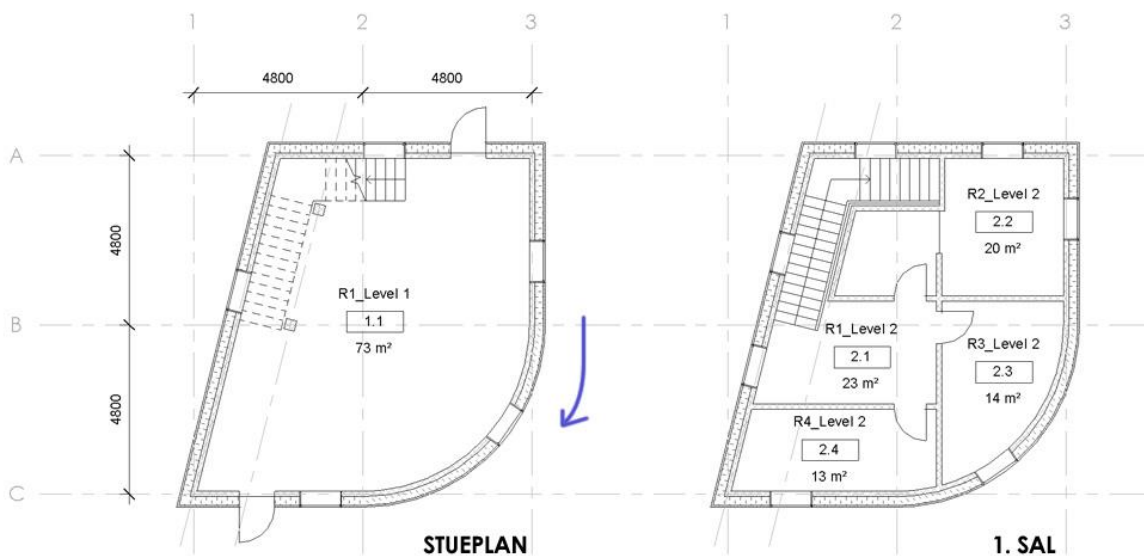
Figur 2.2.2.1. Isometriske skærbilleder af virtuel bygningsmodel

<sup>2</sup> BIM- Platform/Applikation er skildret iht. (Eastman, et al., 2011, pp. 585-586)

Bygningsfakta	
Bruttoareal:	178 m <sup>2</sup>
Etager:	2
Taghældning:	3° ensidigt fald
Rum:	5
Vinduer:	12
Yderdøre:	2
Indvendige døre:	3
Bygningsdele <sup>3</sup>	
Tag (%BE2_590):	Betondæk og isoleringslag med tagpap.
Facader (%AD1 - Ext_450):	Bagmur i beton, isoleringslag, klimaskærm i beton.
Murkroner (%BD1 - Ext_300):	Som facader ekskl. Bagmur.
Indervægge (%BD2 - Int_150):	Beton.
Fundament (%AB1 - Ext_300):	Beton.
Terrændæk (%AC1_450):	Betonplade på isoleringslag.
Etagedæk (%AC2_300):	Beton.
Vinduer (%QA1 - 1200x1200):	Enkeltfag 1200 * 1200 mm.
Yderdøre (%QCC2 - Ext_10x21):	Glas 1000 * 2100 mm.
Indvendige døre (%QCC3 - Int_9x21):	Generisk 900 * 2100 mm.

Tabel 2.2.3. Bygningsfakta om testmodel/sekundær case

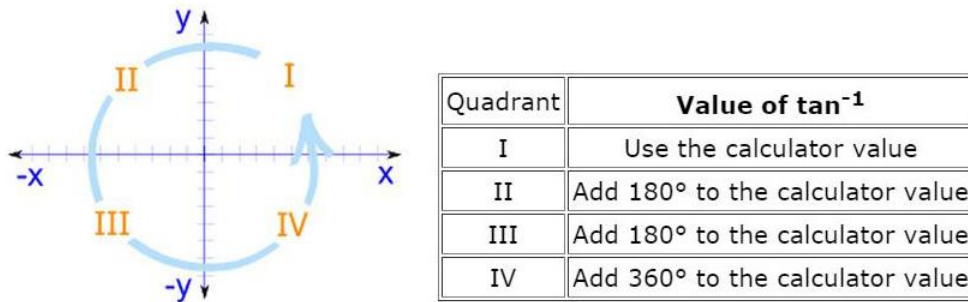
Facaderne er modelleret i med urets retning og omfatter både rette og runde vægge (nedenstående figur). Derudover er bruttoarealet modelleret på hver etage.



Figur 2.2.2.2. Planvisninger af testmodel. Den blå pil illustrerer modelleringsretning

<sup>3</sup> Bygningsdele er navngivet med overordnet CCS (Cuneco Classification System) type-ID, efter cuneco klassifikationstabel Der blev angivet en ugyldig kilde.

Vinduer er placeret i hver af de 4 kvadranter i henhold til nedenstående figur:



Figur 2.2.2.3. Kvadranter og omregning af polære og retvinklede koordinater (MathsFun.com, 2013)

Be10 betragter orientering efter kompasorientering og/eller grader stigende *med* urets retning ved nord svarende til  $0^\circ$  og øst svarende til  $90^\circ$  (Aggerholm & Grau, 2011). Af hensyn til gennemskuelighed og sammenligning af metoder, er orienteringer i de individuelle analyser beregnet efter et matematiske X- og Y-koordinater, hvor X svarer til  $0^\circ$ , stigende *mod* urets retning, som illustreret ovenfor. En beregnet orientering for et givent vindue fungerer derfor udelukkende som referenceværdi og ikke i svarende til den reelle input værdi i Be10.

### 2.2.3 TEST: GREEN BUILDING XML (GBXML)

I det følgende beskrives et konkret forsøg med eksport af testmodellen fra Revit til gbXML-formatet, udvikles målrettet til udveksling af informationer fra BIM-modeller til energianalyseværktøjer.

Eksport til gbXML sker på baggrund af en analytisk model genereret efter en given bygningsmodel i Revit. Den analytiske model til eksport er betegnet Revit *Energy Analytical Model* (EAM). Det er den analytiske model, der eksporteres til gbXML (Autodesk Inc., 2014d). Dette kan som udgangspunkt være problematisk i forhold til testmodellens geometriske udformning:



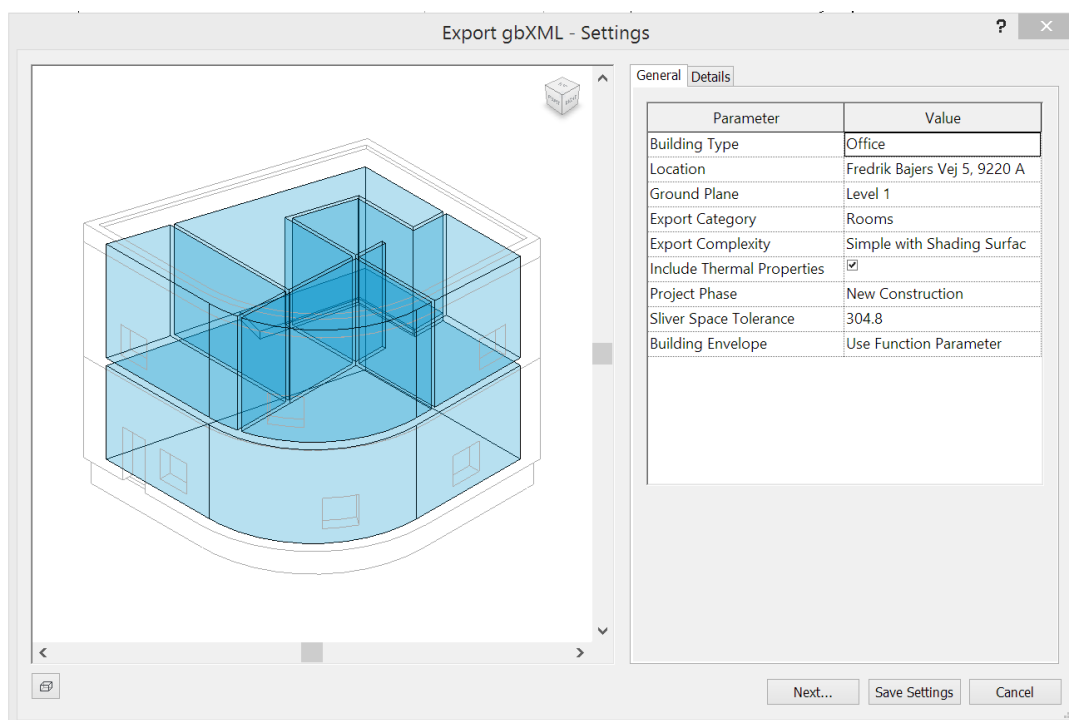
OPENINGS IN CURVED WALLS ARE CURRENTLY UNSUPPORTED

(Autodesk Inc., 2014e)

Hvad det indebærer er ikke uddybet og *preview* i forbindelse med opsætning til gbXML eksport viser ingen umiddelbare problemer. Her har jeg indstillet følgende:

- *Building Type: Office*
- *Include Thermal Properties*
- *Building Envelope: Use Function Parameter*

De resterende indstillinger er standard.



Figur 2.2.3.1. Revit: Preview og indstillinger for gbXML eksport



I dette tilfælde er filen eksporteret med en *encoding*<sup>4</sup> som gør det umuligt at åbne den i DDS-CAD Viewer, hvis ikke det ændres til *UTF-8* i en texteditor som eksempelvis Notepad. Den samme problematik gør sig gældende for Autodesk's analyseværktøj, Ecotect.

### 2.2.3.1 OBSERVATIONER

XML-sproget er relativt læsbart sammenlignet med eksempelvis EXPRESS. I eksemplet herunder har jeg åbnet filen i Notepad++ og fundet information om et vindue.

```
<Opening windowTypeIdRef="aim0914" openingType="OperableWindow" id="aim0966">
  <RectangularGeometry id="aim0967">
    <CartesianPoint>
      <Coordinate>1.2</Coordinate>
      <Coordinate>0.9</Coordinate>
      <Coordinate>0</Coordinate>
    </CartesianPoint>
    <Width>1.2</Width>
    <Height>1.2</Height>
  </RectangularGeometry>
  <PlanarGeometry>
    <PolyLoop>
      <CartesianPoint>// udeladt i eksempel ...
    </CartesianPoint>
  </PolyLoop>
</PlanarGeometry>
<CADObjectId>Sg1: %QA1 - 1200x1200 [337830]</CADObjectId>
<Name>E-11-E-W-2-W-1</Name>
</Opening>
```

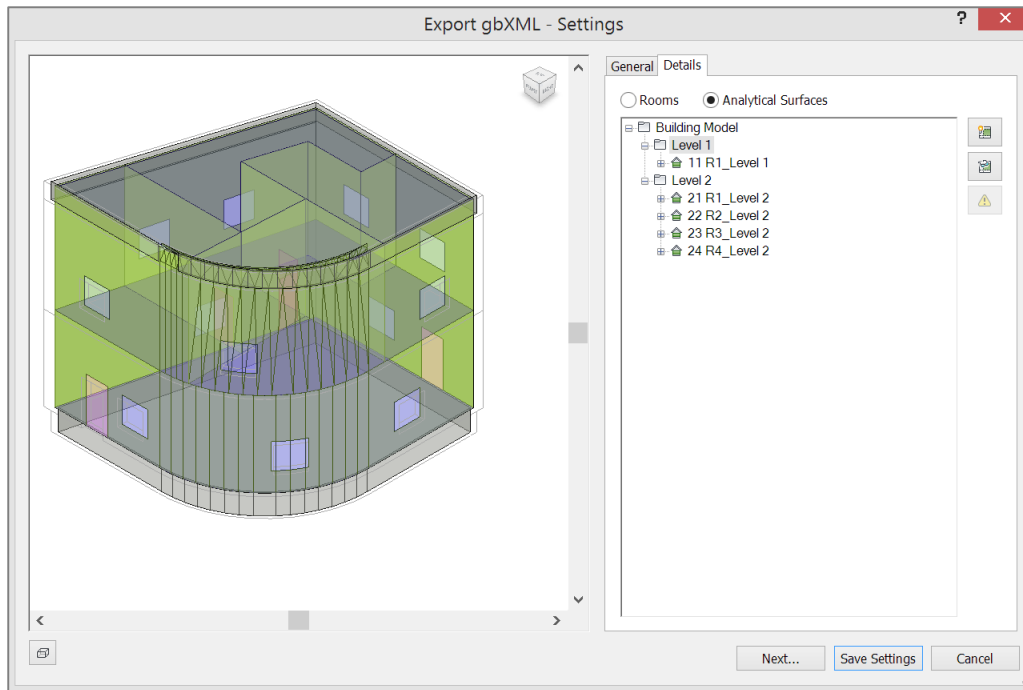
Figur 2.2.3.2. Notepad++: gbXML-model som tekst. Eksempel for vindue (udsnit)

XML filen består af adskillige linjer tekst, men disse få linjer fortæller meget om et enkelt vindue.

### GEOMETRI

Previewet i figur 2.2.3.1 viser ingen umiddelbare problemer i forhold til den runde væg. Dette ændrer sig ved at vælge *'Analytical Surfaces'* frem for *'Rooms'* i eksport indstillingerne. Det ser imidlertid ikke ud til at påvirke, hvordan vinduerne er placeret.

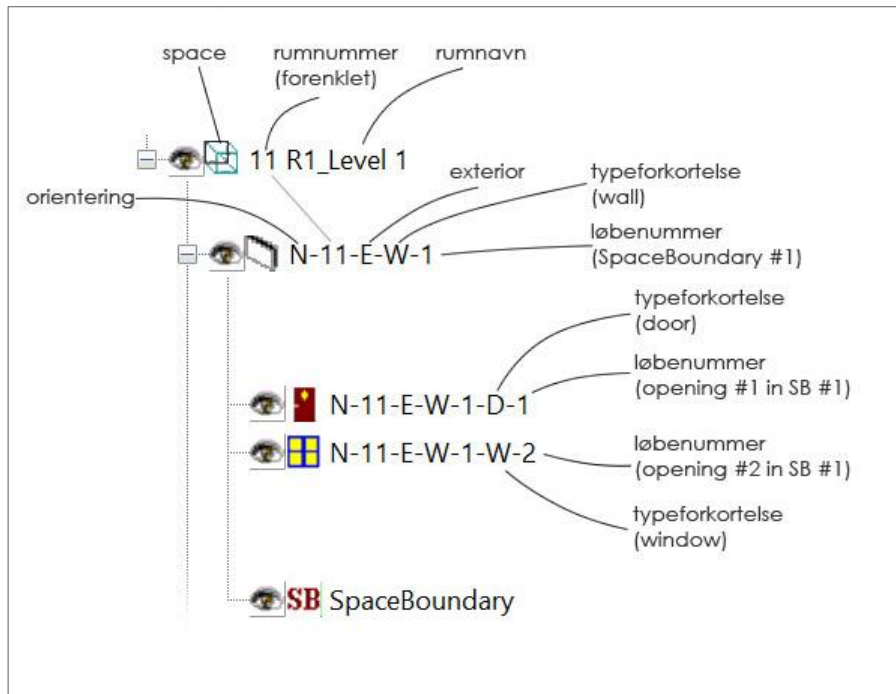
<sup>4</sup> Encoding definerer en sammenhæng mellem data og bytes (van Kesteren, 2014)



Figur 2.2.3.3. Revit: Preview og indstillinger for gbXML eksport (Analytical Surfaces)

## BYGNINGSDELE

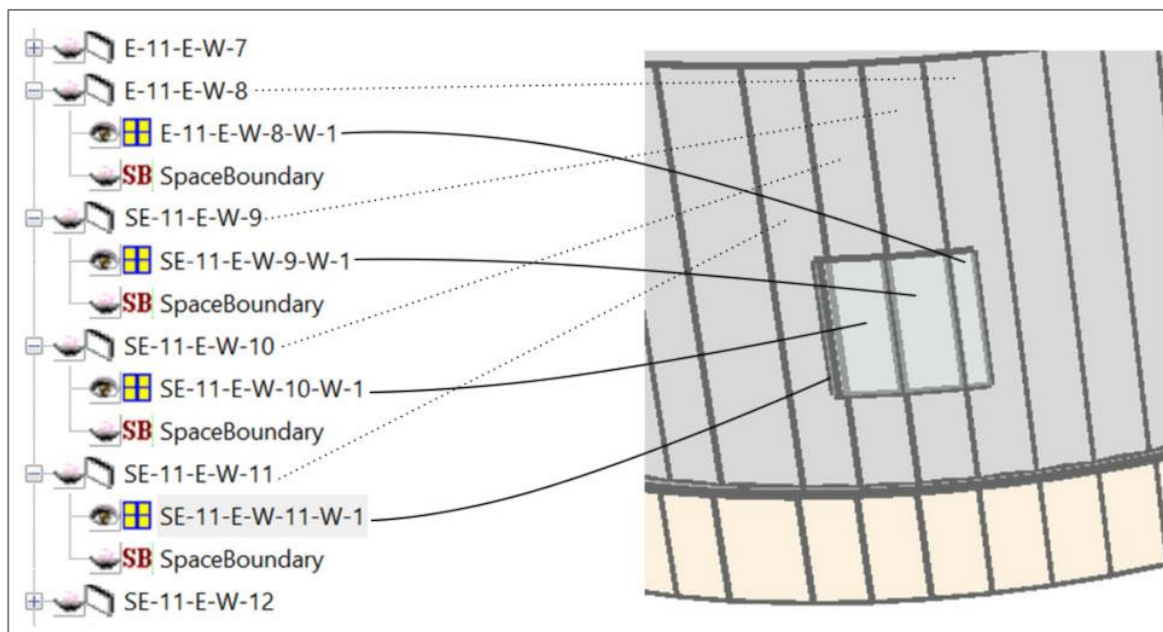
BIM-objekter er navngivet med en funktionskode. Kodestrukturen er bygget op med hvert rum som supertype til de afgrænsende dele, der fungerer som supertype for åbninger, f.eks. vinduer og døre i vægge.



Figur 2.2.3.4. Revit-navngivning af BIM-objekter i testmodel eksporteret til gbXML

Rumkoden består af nummer og navn, uden punktum. Dette nummer nedrives til subtypen væg, som er angivet med en retning som præfiks (f.eks. N). Vægkoden er tilføjet funktions- og typeangivelse (her E og W) samt et løbenummer for *space boundary*. I ovenstående eksempel betegner nummeret første afgrænsende element for rum 11 (nr. 1.1 i Revit). Vægkoden nedrives ligeledes til subtyperne, som ligeledes er angivet med typekode og løbenummer. Vægge der afgrænser flere rum, indeholder koden for hvert af disse rum, eksempelvis N-**21-22**-E-W-15 (Autodesk Inc., 2014d)

Alle bygningsdele er således angivet med en orientering. Det gælder også de runde vægge, hvor konsekvensen af manglende understøttelse i *Revit EAM* er mere fremtrædende.



Figur 2.2.3.5. Opdeling af væg og vindue i eksporteret gbXML-model

De runde vægge er delt op i flere små, hvilket som sådan kan være logisk i forhold til orientering, geometri og fysiske bygninger. Til gengæld arves denne opdeling af vinduet. I eksemplet fremstår det enkelte vindue som 4 individuelle forekomster. Første del er angivet som åbning nr. 1 i væg-space boundary nr. 8 for rum nr. 11. Anden del er angivet som åbning nr. 1 i væg-space boundary nr. 9 for rum nr. 11 og så fremdeles.

### 2.2.3.2 OPSUMMERING AF METODETEST: GBXML

Indledningsvis er problemet med encoding et relativt overkommeligt problem, som ud fra analysen ikke kan udelukkes at opstå i det anvendte setup til forsøget. Det afgørende i forhold til en vurdering af gbXML er Revit Energy Analytical Model (EAM), som genererer modellen til eksport.

Den runde væg skaber flere problemer i den energimodel, der eksporteres til gbXML. Den er således ikke i stand til at håndtere åbninger, som Autodesk gør opmærksom på, men det påvirker imidlertid også tilstødende geometri i forbindelse med taget. Selvom Autodesk gør opmærksom på den manglende understøttelse af runde vægge, er opdelingen af vinduer iøjnefaldende og skaber et falsk billede af den faktiske model. Det er ligeledes u hensigtsmæssigt, at problemer med håndtering af geometri er usynlige indtil '*Analytical Surfaces*' aktiveres frem for standard-previewet, der viser rum. En bruger, der eksporterer en model uden at afprøve forskellige indstillinger, kan således overse problemet indtil den eksporterede model åbnes i et andet værktøj. Hvis en given bruger er uopmærksom på denne problematik, kan fejlsøgning således principielt omfatte to stykker software, frem for det ene, der er kilde til fejl.

I forhold til vinduesorientering, er hvert enkelt objekt beriget med relevante informationer, men som følge af problemet i EAM, er det umuligt at konkludere, hvorvidt selve gbXML-formatet egner sig som informationskilde til Be10, når udgangspunktet er eksport fra Revit. Det er sandsynligvis muligt, at løse den geometriske fejl i samlingen mellem rund facade og tag, men det ændrer ikke på at Revit EAM opdeler vinduer i facaden. Opdelingen alene, gør informationer upålidelige.

Principielt kan gbXML være den bedste af den egnede af nærværende undersøgelses inddragne metoder, for Revit modeller med udelukkende rette vægge, men en sådan forudsætning er uholdbar.

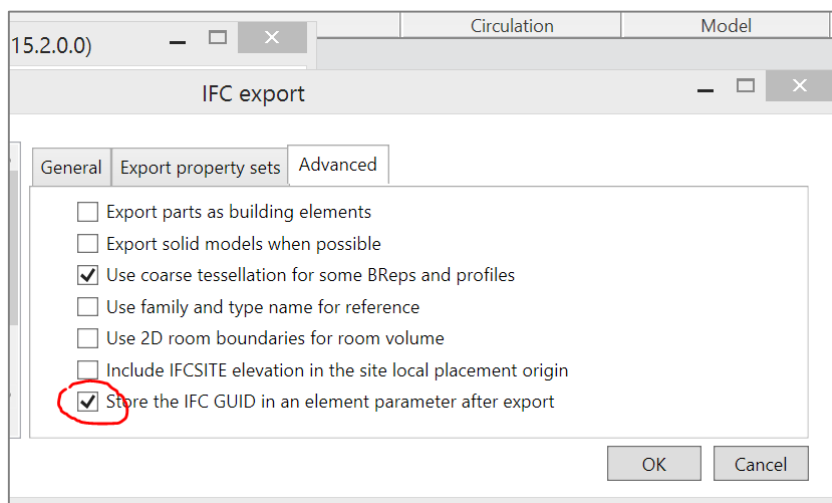
På baggrund af dette vurderer jeg at metoden er irrelevant at tage i betragtning som mulig løsning i det pågældende projekt. I den forbindelse må det samtidig understreges, at denne betragtning er direkte relateret til kompatibilitet i Revit EAM og at analysen således hverken kan bidrage til med argumenter for eller imod gbXML.

## 2.2.4 TEST: INDUSTRY FOUNDATION CLASSES (IFC)

Til opsætning af IFC eksport har jeg anvendt en add-in kaldet *IFC Exporter for Revit 2015* (version 15.2.0.) Den udvider Revit's eksport-funktion og tilføjer en UI med en række supplerende konfigurationsmuligheder. (Velez, 2014)

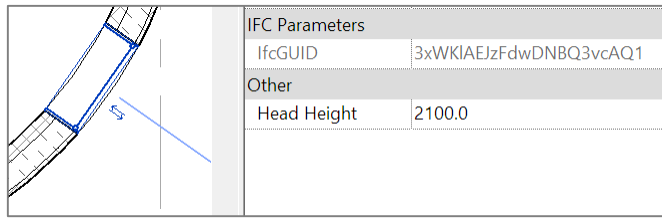
Én af disse muligheder er indstillinger for *Model View Definition (MVD)*. I dette forsøg, har jeg anvendt *Concept Design BIM 2010 (CDB-2010)*. CDB-2010 er overordnet MVD for *Nordic Energy Analysis*, administreret af det norske Statsbygg. (BLIS, 2014)

IFC Exporter tillader en brugertilpasset eksportopsætning med udgangspunkt i en officiel MVD. For at være i stand til at sammenligne IFC-filen med objekter i Revit, har jeg tilføjet en indstilling som vist i figuren herunder.



Figur 2.2.4.1. Brugertilpasning af IFC-eksport i Revit 2015

Denne indstilling sikrer, at objekters *IFC GUID (Globally Unique Identifier)* er; (1) synligt i egenskabsdata i Revit og (2) forbliver uændret selvom modellen eksporteres flere gange.



Figur 2.2.4.2. Revit parameter med IfcGUID

#### 2.2.4.1 OBSERVATIONER

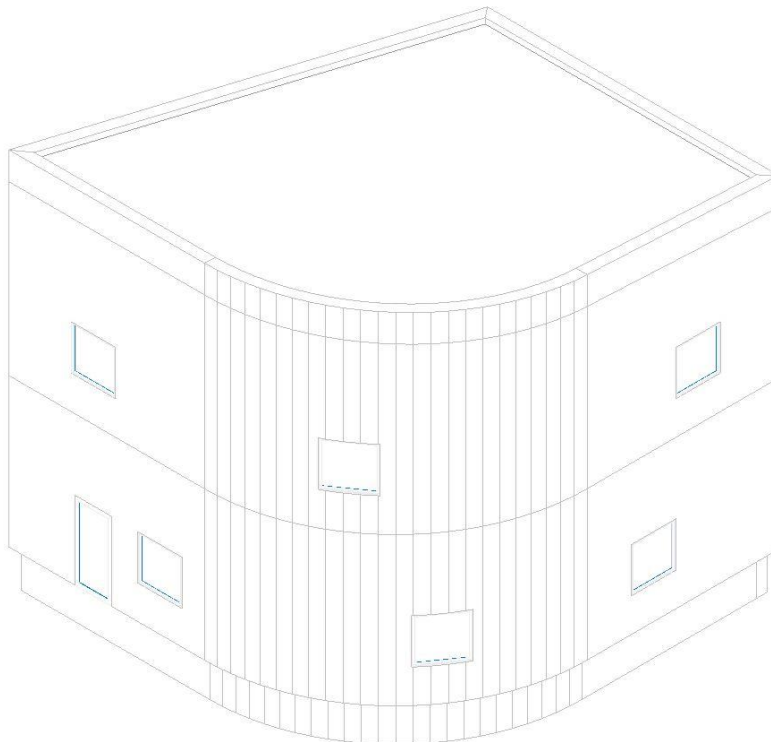
Den eksporterede IFC-model skaber i dette tilfælde ingen problemer i forhold encoding. EXPRESS sproget kan derimod være svært at læse i forhold til eksempelvis XML. Vinduet i ovenstående eksempel, er repræsenteret af følgende i Notepad++:

```
#7471= IFCWINDOW('3xWK1AEJzFdwDNBQ3vcAQ1',#41,'Win_Sgl:%QA1 -
1200x1200:337156',$,'%QA1 - 1200x1200',#17217,#7463,'337156',1200.,1200.);
```

Figur 2.2.4.3. Notepad++: IfcWindow repræsenteret som tekst (oprindeligt én linje)

#### GEOMETRI

DDS-CAD Viewer viser at IFC-modellen er i stand til at håndtere geometrien.

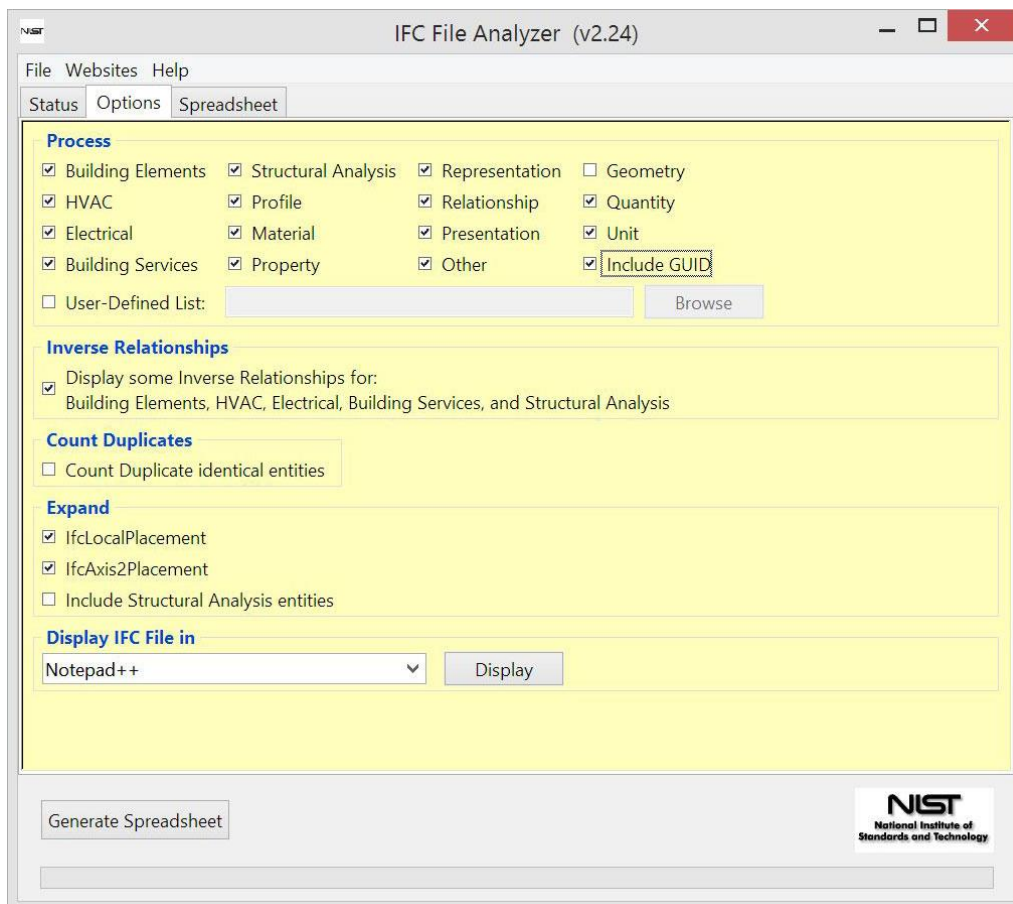


Figur 2.2.4.4. DDS-CAD: Isometrisk visning af testmodel i IFC-format

## IFC FILE ANALYZER

IFC-modellen har jeg analyseret ved hjælp værktøjet *IFC File Analyzer*. Værtøjet er i stand til at generere Excel-regneark fra IFC-filer og sortere data efter entiteter i en model (Lipman, 2014).

Under *Options* findes en række valgmuligheder for typer af entiteter, der skal behandles og skrives ud i regnearket. Informationer om placeringer og retninger medtages ved at vælge *IfcLocalPlacement* og *IfcAxis2Placement*, under *Expand*.



Figur 2.2.4.5. IFC File Analyzer: Indstillinger anvendt i forbindelse med analysen

*IfcLocalPlacement* og *IfcAxis2Placement* er aktiveret i forbindelse med denne analyse, ligesom *Include GUID* er valgt for at kunne sammenligne med Revit. De resterende indstillinger er *default*.

Udviklingen af IFC File Analyzer stoppede efter 2011 (Lipman, 2014), men værktøjet understøtter udover IFC2x3 også IFC4 specifikationen, som blev udgivet i 2013, efter 6 års udvikling (buildingSMART, 2013b).



## OBJEKTER

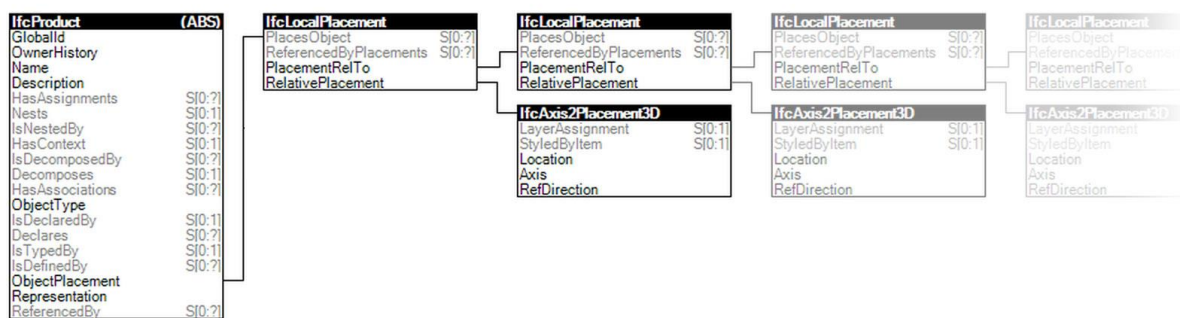
I det regneark IFC File Analyzer har genereret ud fra testmodellen, indikerer oversigten under *summary*, at IFC-modellen håndterer vinduer i de runde vægge efter hensigten. Antallet af vinduer svarer til det antal der er modelleret (12).

	Entity	Count
8		
22	<a href="#">IfcWindow</a>	12

Figur 2.2.4.6. IFC File Analyzer/Excel: Antal af IfcWindow under summary

## PRODUCT PLACEMENT

Entiteter der stammer fra *IfcProduct*, er placeret relativt i forhold til *host*. Et vinduesobjekt er placeret/lokaliseret relativt i forhold til den åbning det sidder i. En åbning i en væg har en relativ placering i forhold til væggen, osv. Placeringen kan derfor fremgå som en reference til en anden entitet i en tekstrepræsentation af en IFC-model (buildingSMART, 2013c).



Figur 2.2.4.7. IfcProduct og relativ placering, princip efter (buildingSMART, 2013c)

Denne 'kæde' af relative placeringer følger *spatial structure*, og indeholder et antal led, svarende til et givent objekts plads i hierarkiet. For et vindue ser dette hierarki ud som følger:

IfcSite

IfcBuilding

IfcBuldingStorey

IfcWallStandardCase

IfcOpeningElement

**IfcWindow**

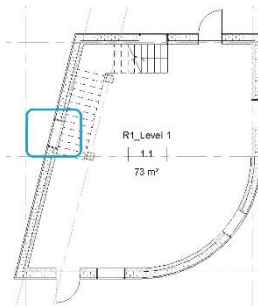
(buildingSMART, 2013d)

## ORIENTERING

En værdi af *PlacementRelTo*, *RelativePlacement* eller *RefDirection* fortæller i sig selv intet konkret, men er således kontekstafhængig. Hvert objekt har et individuelt koordinatsystem, men *RefDirection* kan være defineret forskellige steder og være tilknyttet via en reference. Dette er illustreret i de følgende eksempler, hvor data i et genereret regneark, er sammenholdt med testmodellens geometri i Revit.

I testmodellen er *Site*, *Building* og *BuildingStorey* alle placeret i koordinaterne 0, 0, 0 og derfor udeladt.

## VINDUE I RET FACADE



IfcWallStandardCase:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<7113> IfcLocalPlacement 127	<7113> IfcAxis2Placement3D 7112	-5062.00909 -5229.57052 0.	(Z)	0.24254 0.97014 0.

Tabel 2.2.4.1. IFC File Analyzer/Excel: *ObjectPlacement* for væg

Location definerer den relative placering af væggen nulpunkt, i forhold til etagens nulpunkt. I dette tilfælde er en retning også defineret for væggen (*RefDirection*) med x og y. Lodret hældning er defineret af z. Værdierne kan omregnes til orientering på følgende måde:

$$\text{Punkt: } P_{x,y}(0.24254, 0.97014) = 1. \text{ kvadrant} \Rightarrow +0$$

$$\text{Retning: } \tan^{-1}\left(\frac{0.97014}{0.24254}\right) = 75.96 + 0 + 90 = \mathbf{165.96^\circ}$$

Skæringspunktet for x og y er placeret i kvadrant 1 (deraf +0). Orientering findes ved at lægge yderligere 90 til, da værdierne definerer væggen længderetning.

IfcOpeningElement:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<16999> IfcLocalPlacement 7113	<16999> IfcAxis2Placement3D 16998	5771.41002 -225. 900.		

Tabel 2.2.4.2. IFC File Analyzer/Excel: *ObjectPlacement* for vinduesåbning

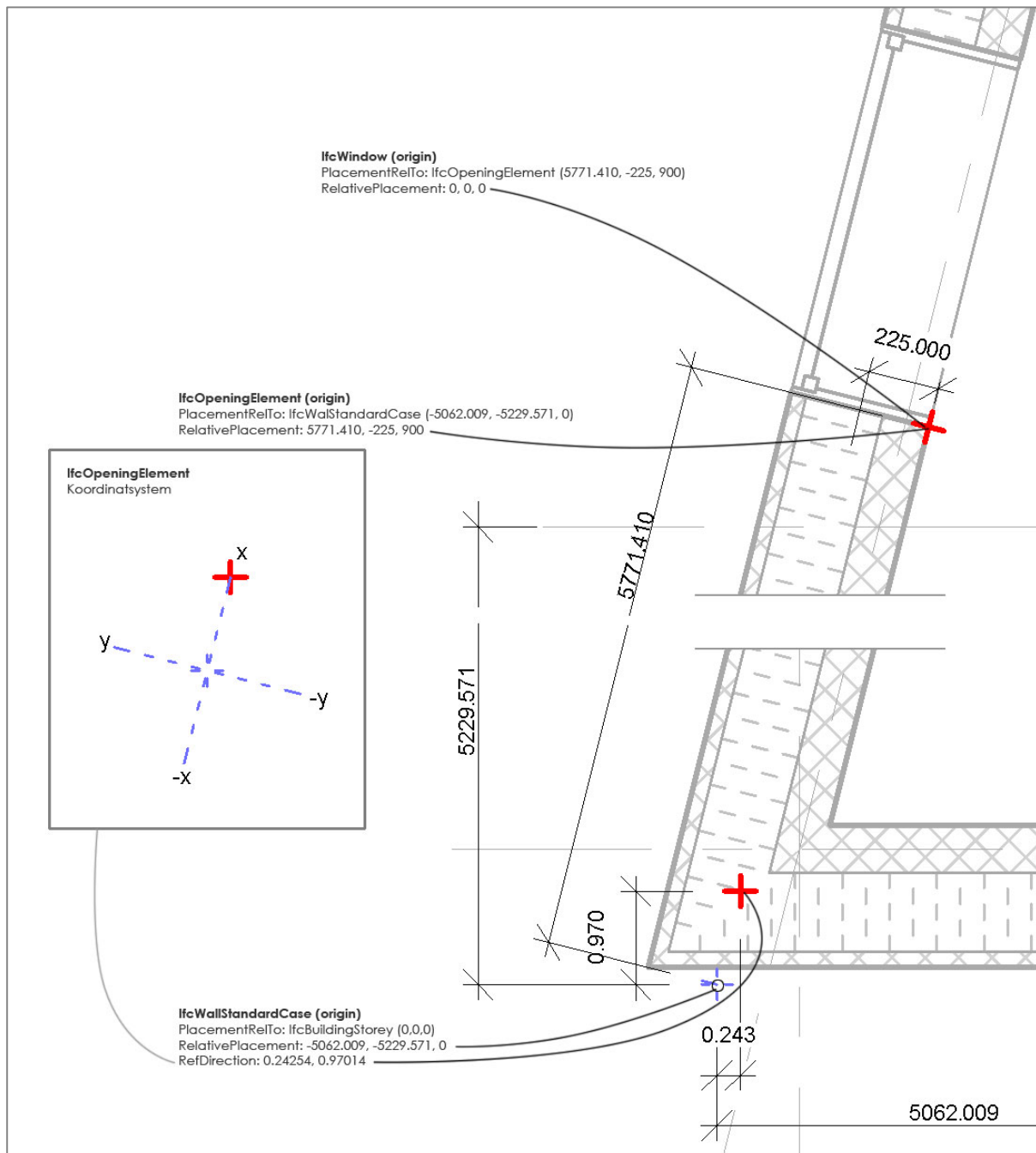
Åbningen har ingen retning defineret. Det skyldes det relative system, der gør væggen længderetning til x-akse i koordinatsystemet for åbningen. Location er således den relative placering af åbningens nulpunkt, i forhold til væggen nulpunkt, hvor z-værdien angiver en brystningshøjde på 900 mm. Den negative y-værdi viser, at længderetningen er placeret i center af væggen, som har en tykkelse på 450 mm. Åbningens nulpunkt er derfor rykket 225 mm mod indvendig side, så den 'skærer' i hele væggen.

IfcWindow:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<17020> IfcLocalPlacement 16999	<17020> IfcAxis2Placement3D 17019	(Origin)		

Tabel 2.2.4.3. IFC File Analyzer/Excel: *ObjectPlacement* for vindue

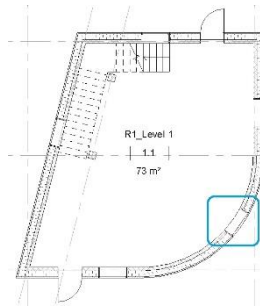
Location for vinduer er angivet som (*Origin*), da vinduerne er modelleret uden fuger og i dette tilfælde har nulpunkt nøjagtigt samme sted som åbningen. De tre objekter og deres respektive værdier er illustreret i nedenstående figur.



Figur 2.2.4.8. Revit: Objektgeometri og skitserede IFC-værdier, lige væg

Figuren illustrerer ligeledes, hvordan væggens længderetning danner grundlag for åbningens koordinatsystem.

VINDUE I RUND FACADE



IfcWallStandardCase:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<7199> IfcLocalPlacement 127	<7199> IfcAxis2Placement3D 7198	4950. 0. 0.		

Tabel 2.2.4.4. IFA/Excel: ObjectPlacement for rund væg

I dette tilfælde er kun *Location* defineret for væggen. Retning er ikke defineret, selvom væggens slutpunkt principielt er placeret i 225°.

IfcOpeningElement:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<17038> IfcLocalPlacement 7199	<17038> IfcAxis2Placement3D 17037	-887.64673 -2112.02563 900.		

Tabel 2.2.4.5. IFA/Excel: ObjectPlacement for åbning i rund væg

Heller ikke åbningen har en retning defineret. Samtidig er nulpunktet forskudt i forhold til væggens geometri (Figur 2.2.4.9).

IfcWindow:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<17063> IfcLocalPlacement 17038	<17063> IfcAxis2Placement3D 17062	120.95339 -84.69247 0.	(Z)	-0.57358 -0.81915 0.

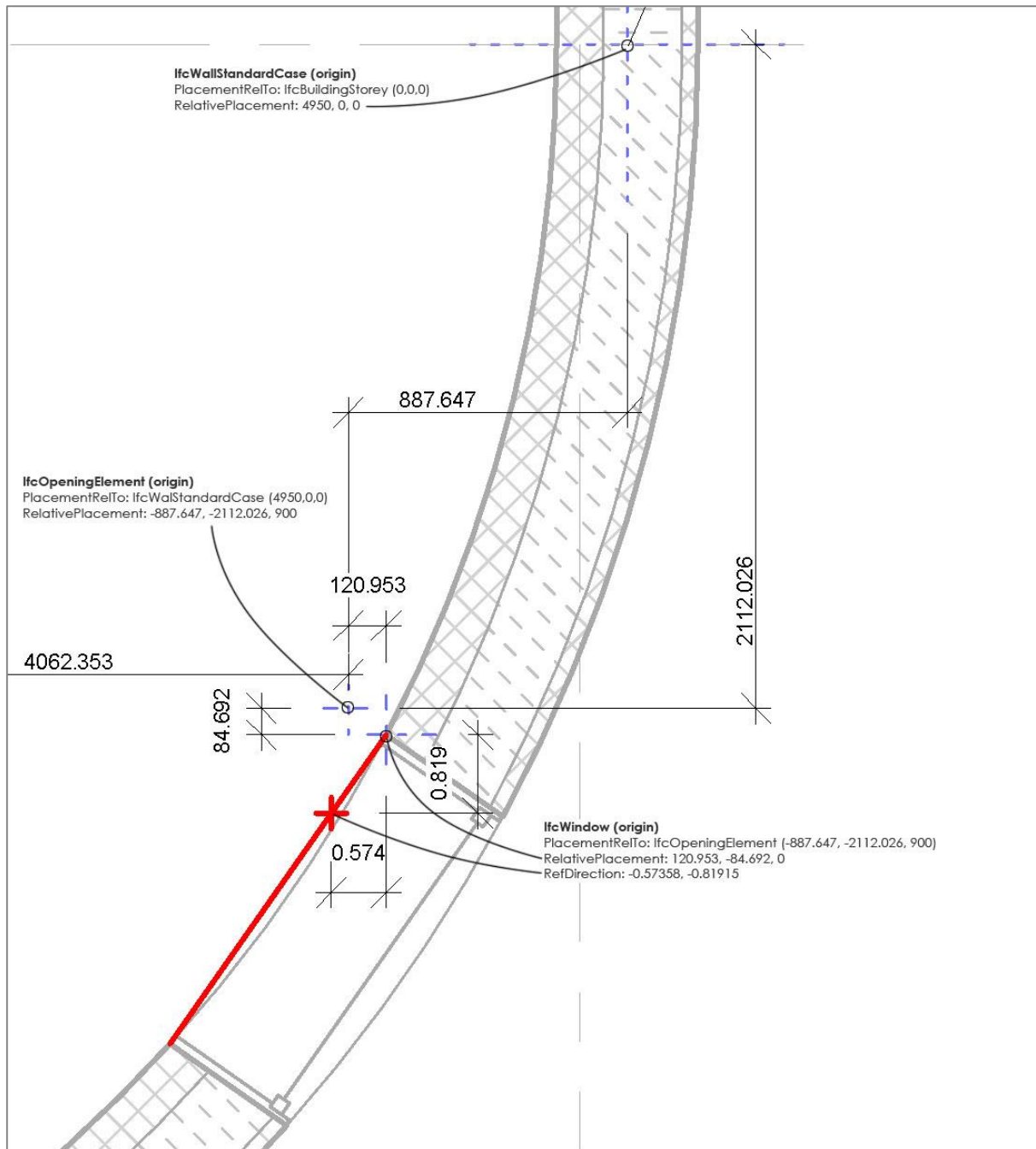
Tabel 2.2.4.6. IFC File Analyzer/Excel: ObjectPlacement for vindue i rund væg

I modsætning til det åbningen, tangerer vinduets nulpunkt geometrien i den runde væg. *Location* er således ikke opgivet som (*Origin*). Hvor retningen var defineret i den lige væg, er den defineret for selve vinduet i den runde væg (*RefDirection*). De negative værdier af x og y, fortæller at skæringspunktet befinder sig i kvadrant 3 (deraf +180). Vinduets orientering kan regnes på følgende måde:

$$Kvadrant: P_{x,y}(-0.57358, -0.81915) = 1. kvadrant \Rightarrow +180$$

$$Retning: \tan^{-1} \left( \frac{-0.81915}{-0.57358} \right) = 55 + 180 \Rightarrow 235 + 90 = 325^\circ$$

Disse værdier er illustreret i sammenhæng med den runde væg i figuren nedenfor.



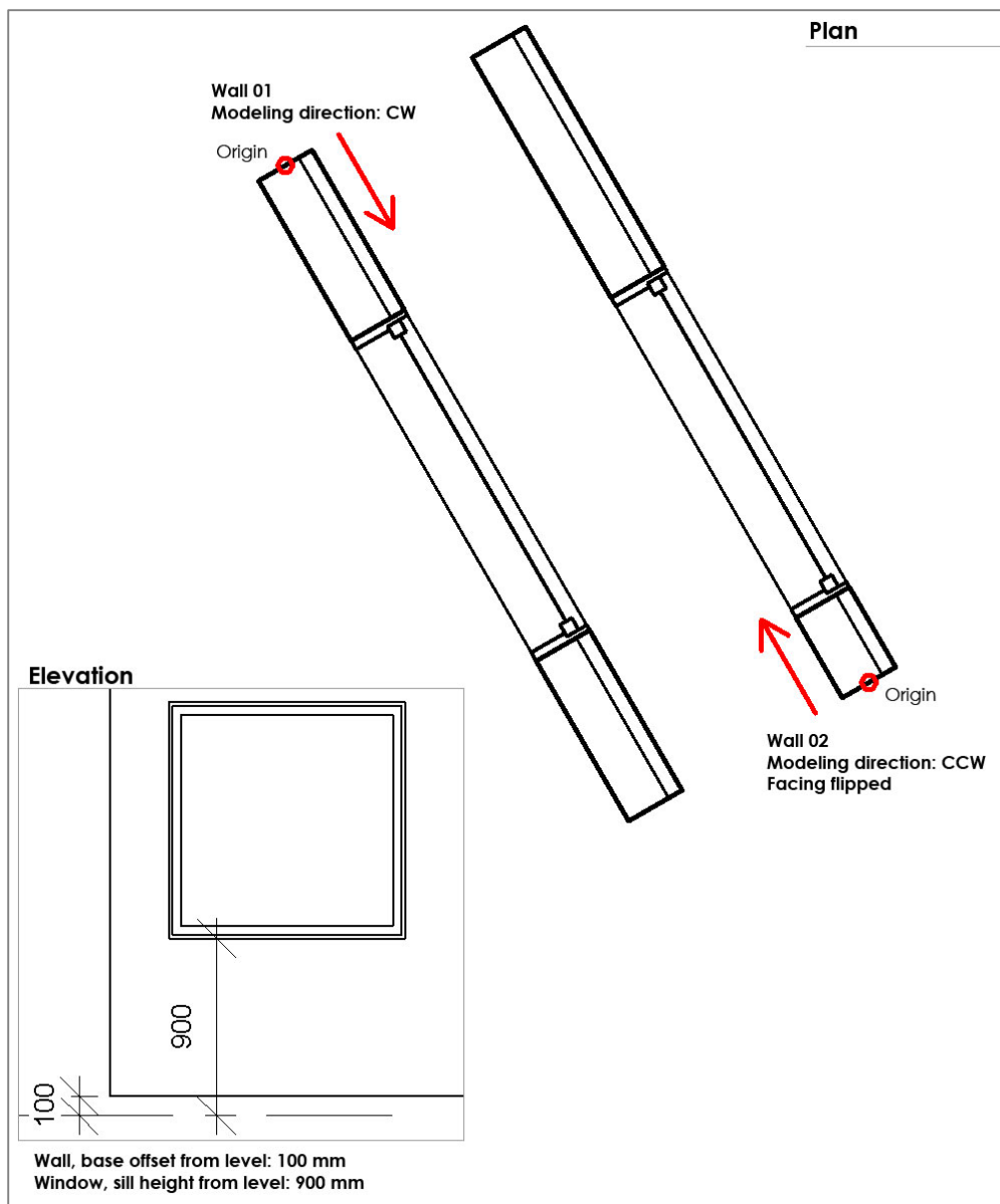
Figur 2.2.4.9. Revit: Objektgeometri og skitserede IFC-værdier, rund væg

I begge tilfælde er åbningens z-værdi angivet til 900, som svarer til den brystningshøjde vinduerne er modelleret med. Selvom *RefDirection* er angivet forskelligt, fremgår værdierne relativt ensartede. Men testmodellen er simpel og facaderne er alle modelleret i urets retning. I et større og mere komplekst projekt, må nogle vægge sandsynligvis modelleres i modsat retning og vendes/flippes efterfølgende.

## MODELLERINGSRETNING

For at undersøge potentielle betydninger af dette, oprettede jeg et isoleret projekt og modellerede 2 vægge med et vindue i hver. I det følgende kaldet *Wall01* og *Wall02*.

Væggene er delt i 2 konstruktionslag med forskellig tykkelse, så det er muligt at identificere den udvendige side visuelt (tyndeste lag).



Figur 2.2.4.10. Modelleringsretning: Wall01 og Wall02

**Wall01** er modelleret oppefra med uret.

**Wall02** er modelleret nedefra mod uret, idet væggen er *flipped*.

Underkant for begge vægge er 100 mm over etage. Vinduerne er orienteret ens har en brystningshøjde på 900mm over etage.

Den isolerede '2-vægs' model er eksporteret til IFC og behandlet med IFC File Analyzer. I dette tilfælde er værdierne i regnearket mere forskelligartede.

## IfcWallStandardCase

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<142> IfcLocalPlacement 127	<142> IfcAxis2Placement3D 141	1200. 600. 100.	(Z)	0.5 -0.86603 0.
<609> IfcLocalPlacement 127	<609> IfcAxis2Placement3D 608	3092.82032 -1078.46097 100.	(Z)	-0.5 0.86603 0.

Tabel 2.2.4.7. IFC File Analyzer/Excel: Wall 01 (øverst) og Wall 02 (nederst)

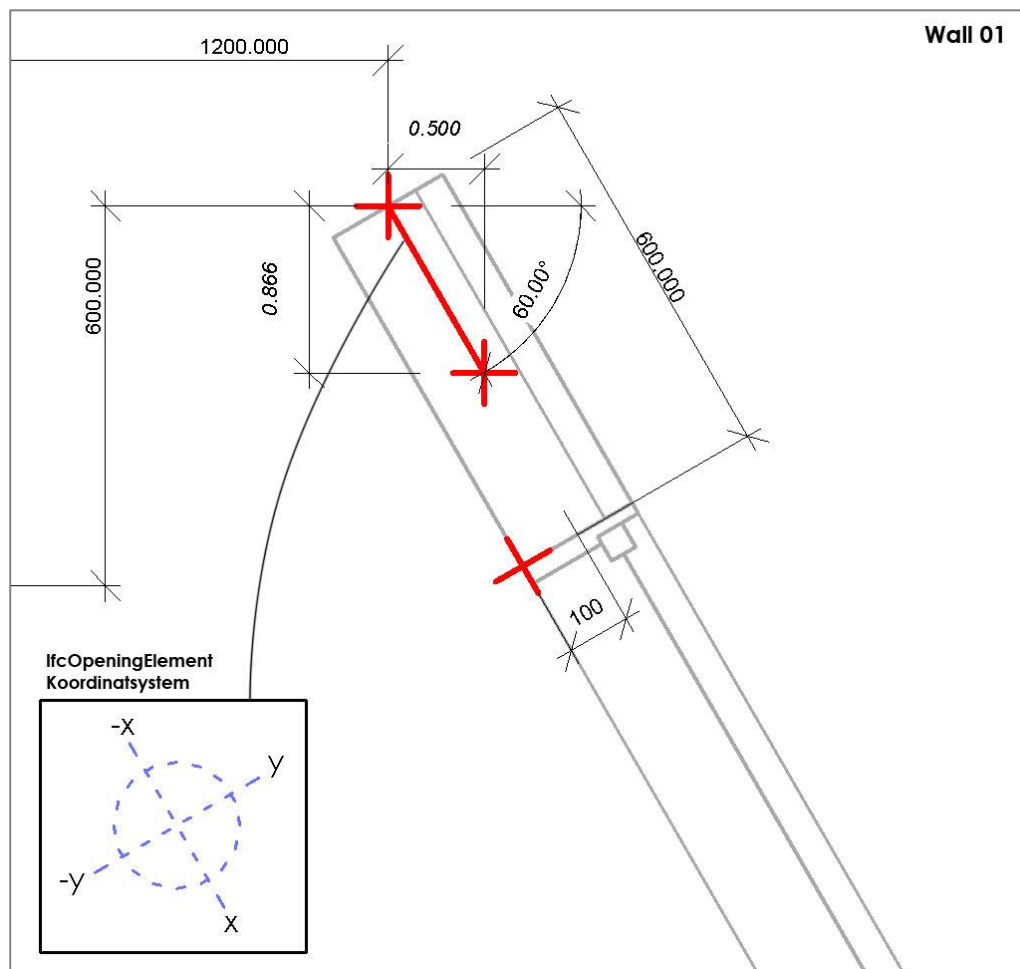
Placering og retning for væggene svarer til den måde de er modelleret på. De udvendige orienteringer fremgår ikke. Afstanden på 100 mm fra underkant væg, til etage er afspejlet i z-værdien.

## IfcOpeningElement

Wall 01:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<925> IfcLocalPlacement 142	<925> IfcAxis2Placement3D 924	600. -100. 800.		

Tabel 2.2.4.8. IFC File Analyzer/Excel: ObjectPlacement for åbning i Wall 01

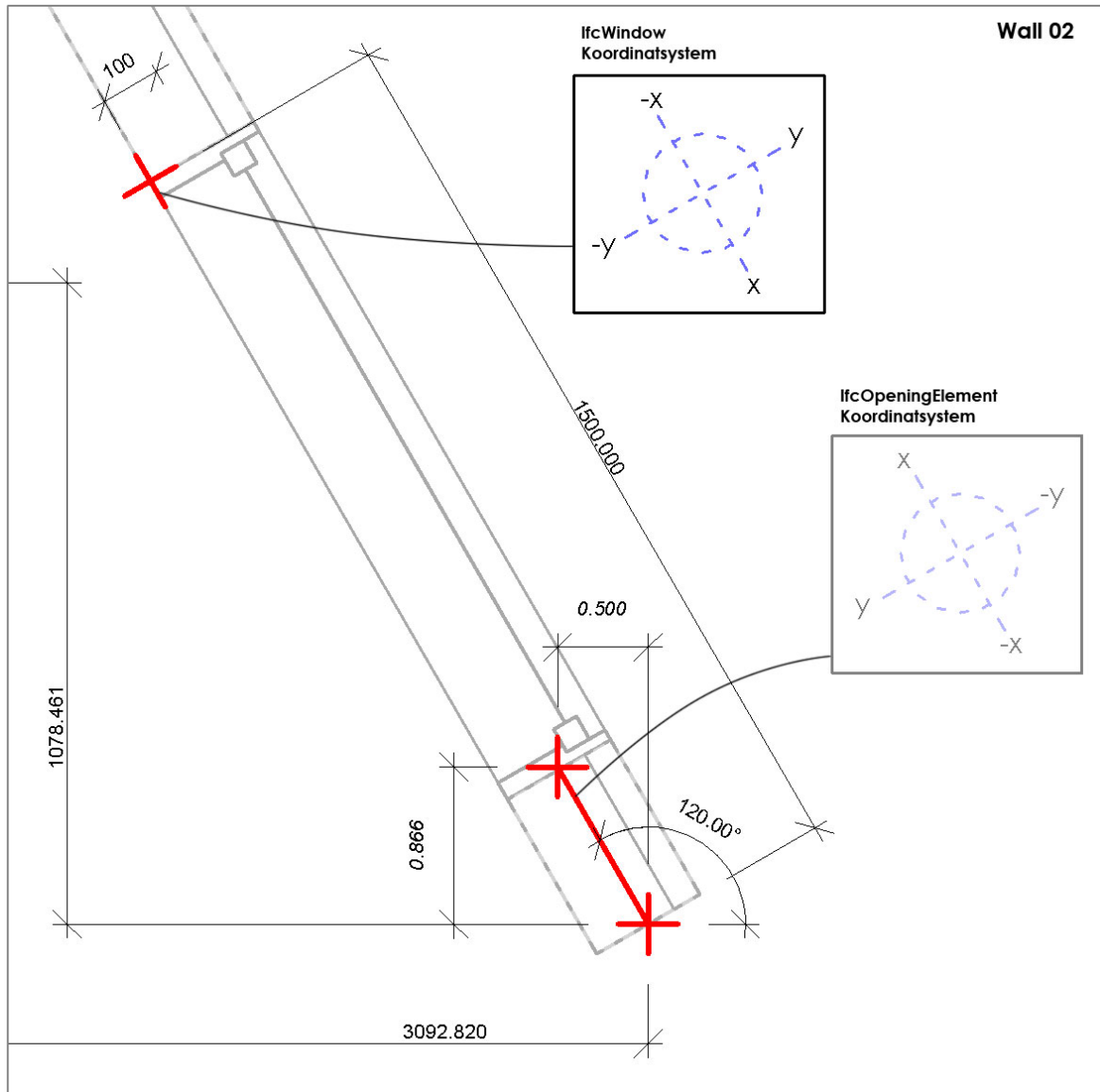


Figur 2.2.4.11. Wall 01: Retning og koordinatsystem for vinduesåbning

Wall 02:

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<964> IfcLocalPlacement 609	<964> IfcAxis2Placement3D 963	1500. 100. 800.		

Tabel 2.2.4.9. IFC File Analyzer/Excel: ObjectPlacement for åbning i Wall 02



Figur 2.2.4.12. Wall 02: Retninger og koordinatsystemer for vindue og åbning

Åbningerne er begge angivet med en z-værdi på 800 og således den relative afstand til underkant væg. Samtidig indikerer x og y værdierne, at koordinatsystemerne er baserede på væggenes modelleringsretninger, uanset om de er *flippet*. Modsat Wall 01, er y-værdien her positiv og nulpunktet for en åbning må således forholde sig til væggenes indvendige side i begge tilfælde.



## IfcWindow

PlacementRelTo	RelativePlacement	Location	Axis	RefDirection
<948> IfcLocalPlacement 925	<948> IfcAxis2Placement3D 947	(Origin)		
<985> IfcLocalPlacement 964	<985> IfcAxis2Placement3D 984	(Origin)	(Z)	(-X)

Figur 2.2.4.10. IFC File Analyzer/Excel: ObjectPlacement for vinduer i Wall 01 (øverst) og Wall 02 (nederst)

Vinduet i Wall 02, er angivet til (-X) i *RefDirection*, hvilket principielt vender koordinatsystemet 180°, så de to vinduer fremstår med samme orientering.

Forskellen på de to scenarier kan visualiseres ved x-værdien, som for Wall 01 angiver afstanden fra væggens nulpunkt til nærmeste side i åbningen (Figur IFA10) For Wall 02, er det modsat og x-værdien angiver afstanden til den fjerneste side i åbningen (Figur IFA11)

De to vinduers orientering kan beregnes på følgende måde:

Vindue, Wall 01:

$$\text{Punkt: } P_{x,y}(0.5, -0.86603) = 4. \text{ kvadrant} \Rightarrow +360$$

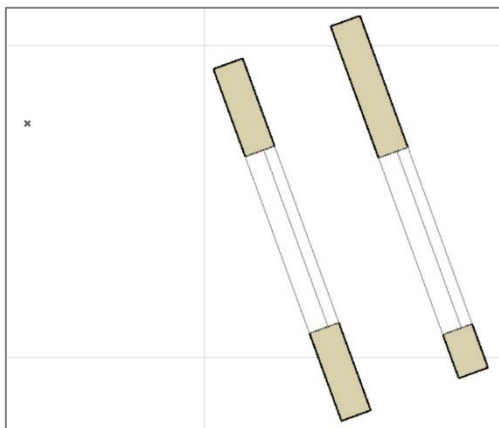
$$\text{Retning: } \tan^{-1}\left(\frac{-0.86603}{0.5}\right) = -60 + 360 \Rightarrow 300 + 90 = 30^\circ$$

Vindue, Wall 02:

$$\text{Punkt: } P_{x,y}(-0.5, 0.86603) = 2. \text{ kvadrant} \Rightarrow +180$$

$$\text{Retning: } \tan^{-1}\left(\frac{0.86603}{-0.5}\right) = -60 + 180 \Rightarrow 120 - 90 = 30^\circ$$

Ved at tage højde for (-X) i beregningerne, går tallene op i sidste ende og tolkes uden synlige problemer i eksempelvis ArchiCAD18.



Figur 2.2.4.13. IFC filen åbnet i ArchiCAD18

### 2.2.4.2 OPSUMMERING AF METODETEST: IFC

Informationer der findes i IFC-modellen er gennem det åbne format tilgængelige og til en vis grad også læsbare ved at følge objekters individuelle referencer til hinanden. Dette gælder ligeledes den definerede geometri og herunder vinduer og orienteringer. De analyserede data omfatter en række eksempler på det relative koncept for objekters placering i modelkonteksten og hvordan det i sammenhæng med objektreferencer kan være betydende for lagring af en specifik værdi. I dette tilfælde har jeg været i stand til at konstatere definerede retninger lagret i de 3 objekttyper, ovenstående beskrivelse fokuserer på. I hvert tilfælde er informationen tiknyttet et logisk objekt, sammenholdt med hierarkiet i *Spatial Structure* og relative placering. XYZ-værdier er således defineret hvor det er nødvendigt og ingen andre steder. Det er testen med de to enkeltstående eksempler på.

Selvom IFC-modellen som udgangspunkt kan være ulogisk og svær at læse for mennesker, kan objektrelationer og referencer være medvirkende til at reducere størrelsen af en IFC-fil og dermed mængden af data, der er nødvendig at behandle i forbindelse med udtræk af eksempelvis orienteringer.

Interoperabilitet mellem IFC-model og Excel er i forvejen håndteret af IFC File Analyzer og programmet tilbyder et vist omfang af filtrering, inden data genereres. Selvom udvikling af programmet er ophørt, kan det med understøttelsen af IFC4, sandsynligvis være kompatibelt med IFC-modeller i en årrække fremover. Med virksomhedens udvikling af et link mellem Excel og Be10, kan vejen fra IFC-model til Be10-beregning være et spørgsmål om at opsætte korrekte mellemregninger og sortere data i Excel. I den forbindelse er de genererede Excel-ark imidlertid præget af den samme problematik som Revit-skemaer eksporteret med NTI Tools, hvor tal er formateret som tekst (beskrevet i forbindelse med procesanalyse, afsnit 2.1.3). I forhold til omformatering er det ligeledes uhensigtsmæssigt, at eksempelvis XYZ-værdier er udfyldt i én celle i det regneark IFC File Analyzer genererer.

I analysen af IFC har jeg fokuseret på entities som *Location* og *ObjectPlacement*, som i dette tilfælde definerer vægges centerlinje. I henhold til de øvrige Be10-parametre er dette utilstrækkeligt i forhold til arealer, der i Be10 måles som brutto (Aggerholm & Grau, 2011). Eventuelt definerede space boundaries kan derfor være forbundet med udtræk fra andre dele af IFC-modellen. Disse data er ikke analyseret i denne undersøgelse, hvorfor jeg således ikke er i stand til at redegøre for lokalisering og strukturer i forhold til informationsudtræk.

## 2.2.5 TEST: REVIT PLATFORM API

### REVIT: MODEL VS PROJEKT

Indledningsvis er det relevant at understrege, at begreberne *Model* og *Projekt*, kan have forskellig betydning i Revit-sammenhæng.

*Model* kan siges at omfatte den repræsentationen af data som en virtuel bygning, med de objekter og informationer, der indgår i den - i princippet de informationer, der kan eksporteres.

*Projekt* kan siges at omfatte indstillinger, skabeloner, dokumentation, etc. og som samtidigt kan indeholde én eller flere modeller. F.eks. kan et Projekt indeholde flere forskellige vægtyper, hvoraf blot få eller en enkelt, reelt er taget i anvendelse.

### TILGÆNGELIGHED

Revit 2015 API er tilgængelig via programmeringssprog, der er kompatible med *Microsoft .NET Framework 4.5*<sup>5</sup>. Herunder *Visual C#, VB.NET* og *C++* (Autodesk Inc., 2014j).

API'en opdateres i takt med opdateringer af platformen eller udgivelser af nye versioner. Disse kan medføre, at specifikke funktioner tilføjes, ændres, udskiftes eller udgår. De konkrete ændringer for en given version fremgår af en '*API Changes and Additions*'-beskrivelse, som følger med i dokumentationen til det pågældende SDK (Autodesk Inc., 2014h).

Revit lagrer alle mål i den interne målenhed, *feet*. Et projekt med andre målenheder beskrives af omregnede værdier, som i API'en fremgår som tekst (Gonçalves, 2014).

API'en er usynlig i Revit *OOTB*<sup>6</sup>. Det er derfor nødvendigt, at foretage en indledende konfiguration for at tilgå informationer, men det er muligt få indsigt i API'en og dens funktionalitet, uden reel programmering. Autodesk stiller en række værktøjer til rådighed i *Revit SDK (Software Development Kit)* (Autodesk Inc., 2014i). Det indeholder en lang række *samples* (eksempler med kildekode). Med en installation af *Revit SDK*, er det muligt at kompilere og afprøve de forskellige samples, som plugins i Revit. Installations- og opsætningsproceduren med Microsoft Visual Studio 2012, er nærmere beskrevet i Appendiks E.

### RVTSAMPLES

*SDK-Samples* giver illustrerer et bredt spektrum af forskelligartede funktioner, der vidner om et stort potentiale for problemløsning og procesoptimering med API-udvikling. Eksempler på relevante samples, kan omfatte:

- *FindSouthFacingWindows* (Markerer alle sydvendte vinduer i model)
- *FireRating* (Parameterstyring, Excel eksport/import af værdier)

En test af *FireRating* samplet er beskrevet i Appendiks F.

<sup>5</sup> Integreret komponent i Windows, der understøtter udvikling med objektorienteret programmering

<sup>6</sup> Out Of The Box: Udtryk for standard udgave/installation af system/software

### 2.2.5.1 REVIT LOOKUP

Informationer og parametre der er tilgængelige i databasen for et konkret Revit-projekt, gennem API'en, er mulige at identificere med udviklerværktøjet *Revit Lookup*.

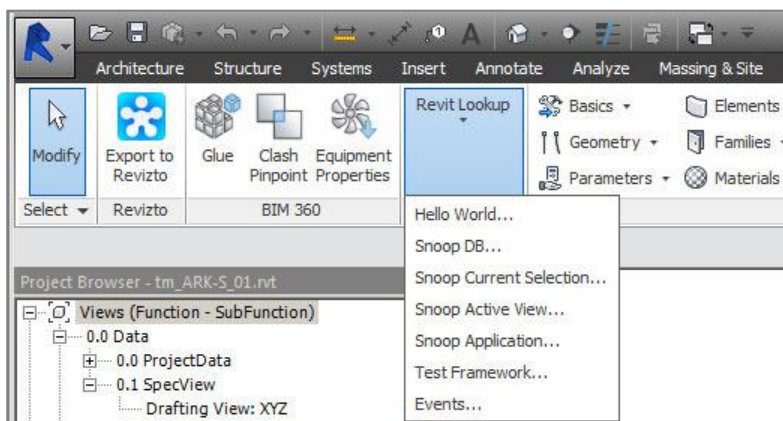


*Dette værktøj gør det muligt at inspicere databasen fra et API perspektiv. Alle informationer der kan vises i værktøjet, er tilgængelige gennem API'en*

(Gonçalves, 2014)

*Revit Lookup* var tidligere en del af *Revit SDK*, men udvikles nu separat (Tammik, 2014a).

Efter installationen af værktøjet, er det tilgængeligt i Revit's UI (*User Interface*) under *Add-Ins* > *Revit Lookup*.



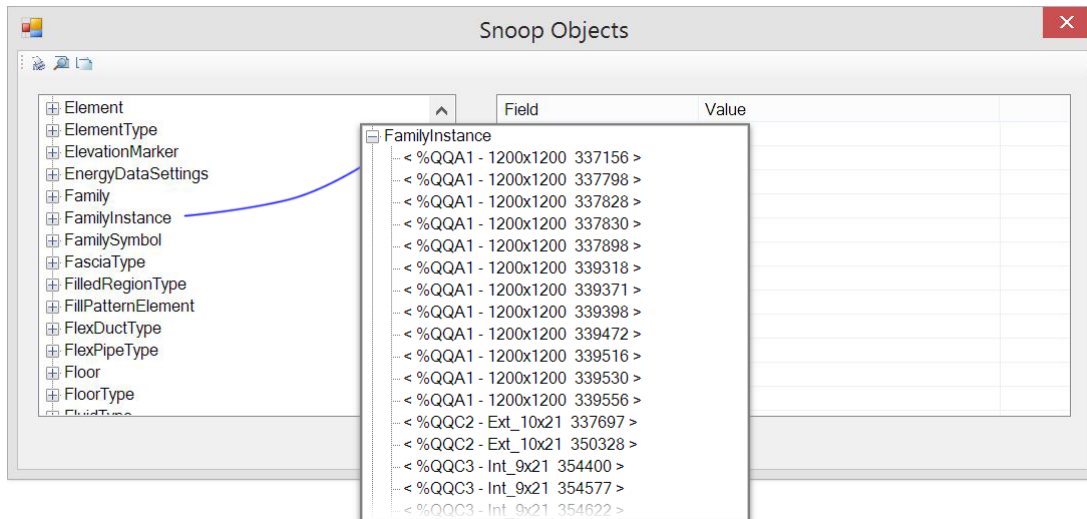
Figur 2.2.5.1. Revit Lookup Add-In i User Interface

### 2.2.5.2 OBSERVATIONER

Revit Lookup kan opliste *Classes* fra eksempelvis *Autodesk.Revit.DB* namespace med relationer og tilhørende informationer. Det kan være som en komplet samling eller tilgængelige informationer for et eller flere specifikke objekter.

#### SNOOP DB

Funktionen '*Snoop DB*' opstiller de *Classes* der indgår i Revit-projektet (families, system families, annotations, etc.) Det omfatter også elementer, som ikke i brug. Listen repræsenterer den samlede mængde information inddelt efter *Class*. Vinduer og døre er oplistet under *FamilyInstance*:



Figur 2.2.5.2. Revit Lookup: Snoop DB. Vinduer/døre under *FamilyInstance*

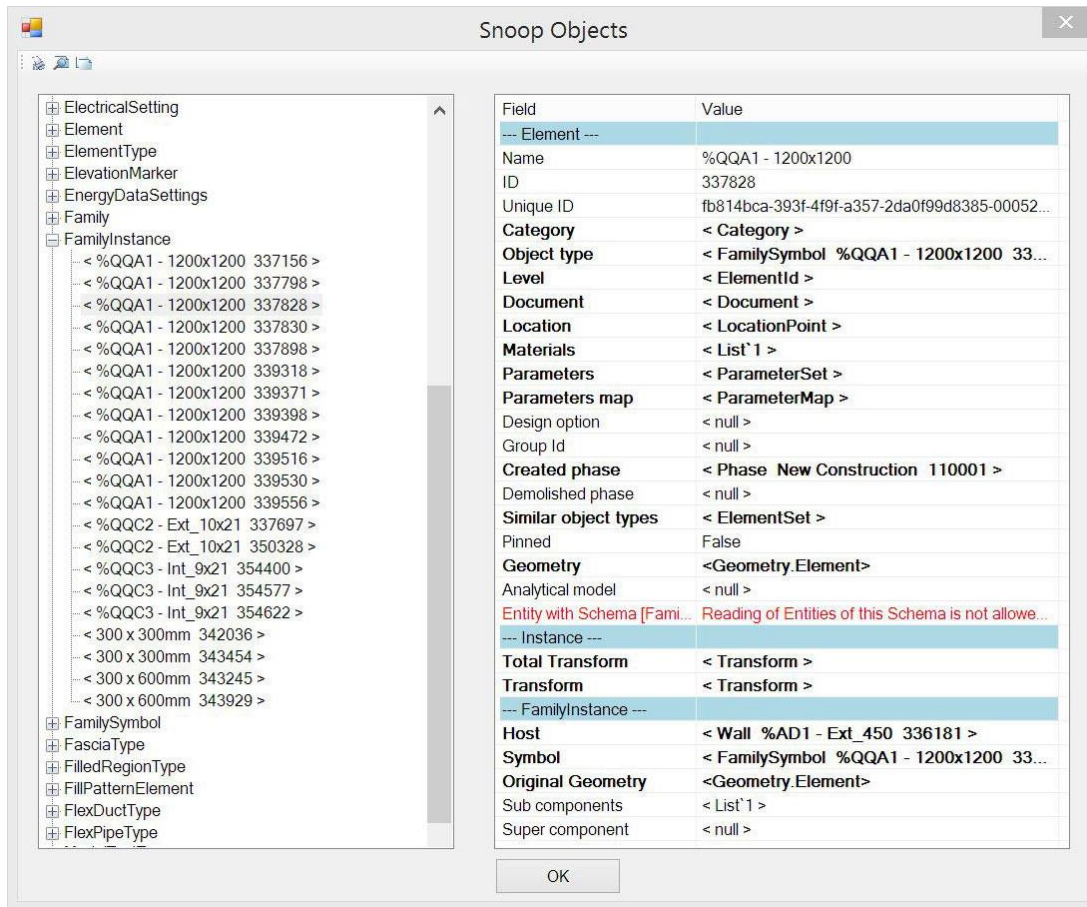
Det er muligt at markere, hver enkelt *FamilyInstance*, eller forekomst, og få vist en række supplerende informationer, som eksempelvis *ID*, *Location*, *Host* og *From room*. Sidstnævnte beskriver det tilstødende rum på vinduets indvendige side. Rumrelationerne omfatter også '*To room*', der i denne bygningsmodel dog kun gør sig gældende for indvendige døre.

Selv de '*tags*' jeg har tilføjet som krydsreference i forbindelse med analysen af Rockwool Energy Design (afsnit 2.2.6), er tilgængelige og beskriver hvilket vindue, hvert enkelt '*tag*' er tilknyttet.

IndependentTag	---	IndependentTag ---
< Mark 408251 >		Leader elbow
< Mark 408259 >		Leader end
< Mark 408266 >		Tag orientation
< Mark 408271 >		Tag head position
< Mark 408506 >		TaggedLocalElementId
< Mark 408508 >		Tag text
< Mark 408512 >		

Figur 2.2.5.3. Snoop DB: 'Mark'-tag og relateret vindue

Informationer forbundet med vinduer er inddelt under 3 overordnede kategorier. Felterne svarer til 3 *Classes* i Autodesk.Revit.DB-namespace, henholdsvis *Element*, *Instance* og *FamilyInstance*.



Figur 2.2.5.4. Snoop DB: Informationer for vindue (FamilyInstance)

Hvert række under de 3, stammer fra *Members* af den pågældende *Class*. Det omfatter både *Methods* og *Properties*. Flere *Members* af *FamilyInstance* arver værdier fra *Element* og er således identiske i listen. Felter med **fed skrift** repræsenterer en *data type* i form af specifikke *Classes* og indikerer at data er aggregeret eller arvet. *Classes* er markeret som "**< Navn >**". Data fra et geometrisk Revit-objekt, kan således trækkes ud ved hjælp af *Methods/Properties* i forskellige *Classes*. Eksempelvis består *Location* både af *Point* (X, Y, Z-koordinater) og *Rotation*, ligesom *Parameters* er samlet af funktioner fra *ParameterSet* (Autodesk Inc., 2014h).

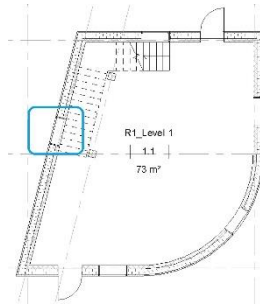
Under *Element*> *Parameters*, er indbyggede Revit-parametre tilgængelige. Disse omfatter blandt andet end areal-parameter, betegnet *HOST\_AREA\_COMPUTED*. Det er imidlertid utydeligt, hvad denne værdi reelt bygger på, idet testmodellens vinduer, der har et overfladeareal på 1.44m<sup>2</sup> med dimensionerne 1200x1200mm, er opgivet til 42.17618. Tallet er jf. (Gonçalves, 2014) angivet i den interne måleenhed *ft*, hvor 42.18 ft<sup>2</sup> omregnes til 3.92 m<sup>2</sup>. Et tal der er væsentligt højere end vinduets areal, selvom både indvendig og udvendig side medregnes.

## ORIENTERING

Informationer kan tilgås i en mere afgrænset form med *'Snoop Current Selection'*. Funktionen viser udelukkende information om det eller de objekter, der er valgt i projektet. Med denne funktion har jeg været i stand til, at gennemgå data relateret til de to vinduer jeg hidtil har fokuseret på, uden at skulle identificere dem først.

Orienteringsværdier er betragtet som almindeligt anvendt information i Revit API, som derfor gør en retningsvektor tilgængelig i egenskaben *FacingOrientation* (Conover, 2010).

## VINDUE I RET FACADE

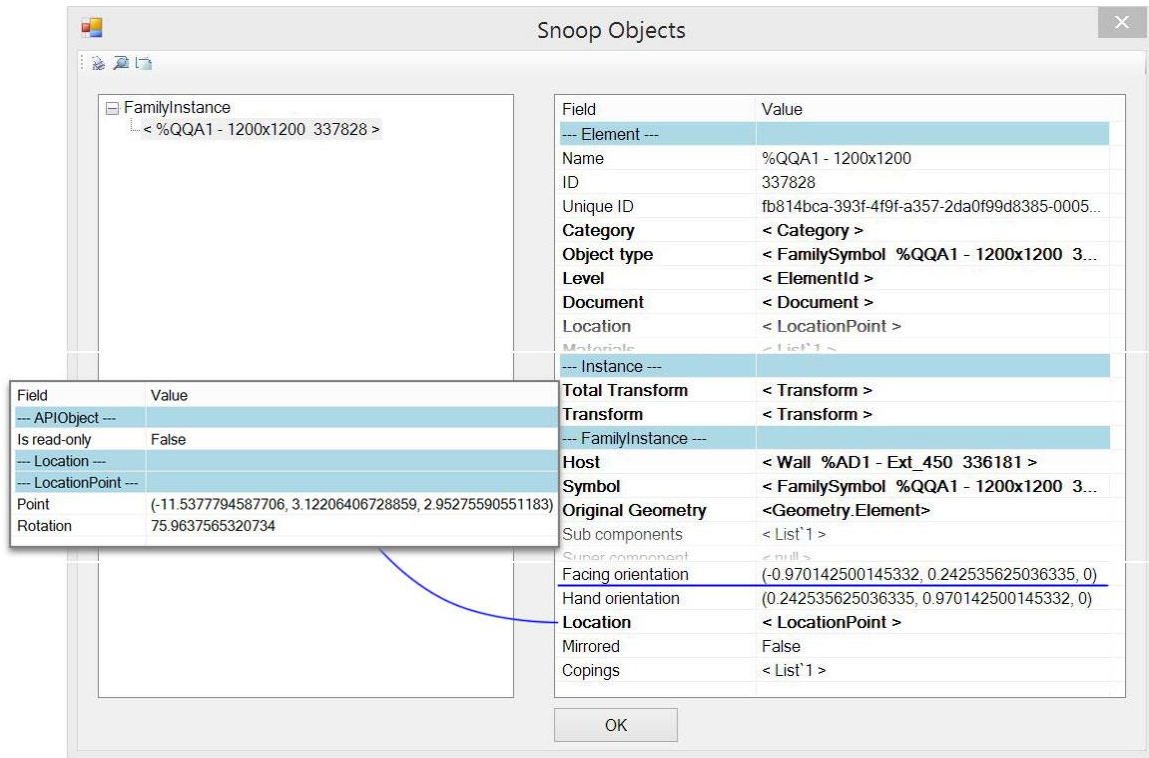


For at sammenligne med IFC, har jeg indledningsvis markeret vinduet i den vestvendte facade og aktiveret *'Snoop Current Selection'*. Med denne funktion viser listen kun den eller de, Classes, som det eller de, valgte objekter tilhører. I dette tilfælde *FamilyInstance* og herunder det markerede vinduesobjekt.

Field	Value
--- Element ---	
Name	%QA1 - 1200x1200
ID	337828
Unique ID	fb814bca-393f-4f9f-a357-2da0f99d8385-000527a4
Category	< Category >
Object type	< FamilySymbol %QA1 - 1200x1200 337138 >
Level	< ElementId >
Document	< Document >
Location	< LocationPoint >
Materials	< List' 1 >
Parameters	< ParameterSet >
Parameters map	< ParameterMap >
Design option	< null >
Group Id	< null >
Created phase	< Phase New Construction 110001 >
Demolished phase	< null >
Similar object types	< ElementSet >
Pinned	False
Geometry	<Geometry.Element>
Analytical model	< null >
Entity with Schema [Fami...	Reading of Entities of this Schema is not allowed to th.

Figur 2.2.5.5. Snoop Current: Vindue (Element Class)

Selvom *FamilyInstance* arver en lang række værdier fra *Element*, indeholder den yderligere informationer om vinduets retning og rumrelationer, som er relevante i forhold til en energianalyse.



Figur 2.2.5.6. Snoop Current: Vindue i ret facade (FamilyInstance / LocationPoint)

Den enkelte forekomst definerer en udvendig retningsvektor og en åbningsretning, med X, Y, Z-værdier i henholdsvis 'Facing orientation' og 'Hand orientation'.

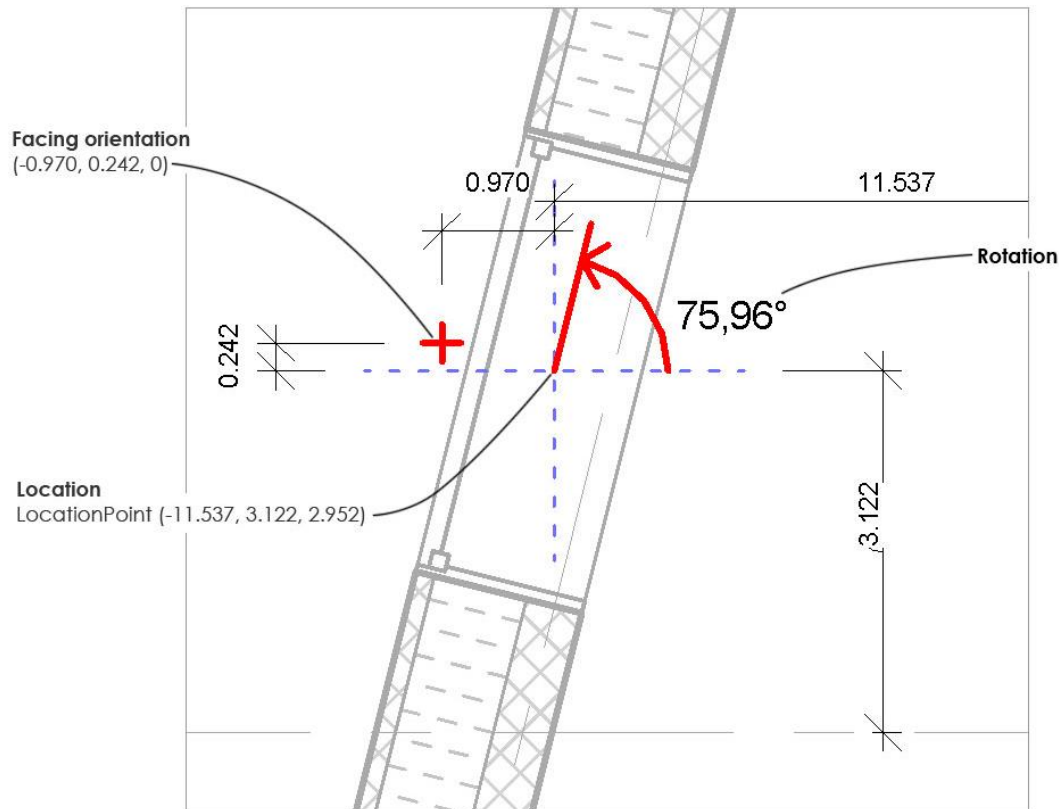
I figur nedenstående har jeg illustreret en række værdier for Location og Facing orientation i sammenhæng med det pågældende vindue.



De stiplede linjer har principielt, hver to funktioner.

- Den horisontale:
  - Repræsenterer **Y**-koordinat for *Location* som afstand til modellens nulpunkt.
  - Symboliserer **X**-aksen i koordinatsystemet for vinduet i modelkontekst.
- Den vertikale:
  - Repræsenterer **X**-koordinat for *Location* som afstand til modellens nulpunkt.
  - Symboliserer **Y**-aksen i koordinatsystemet for vinduet i modelkontekst.

De fuldt optrukne linjer angiver henholdsvis *Facing orientation* og *Rotation*.



Figur 2.2.5.7. Revit: Objektgeometri og skitserede værdier (*Location*, *Rotation*, *Facing orientation*)

Skæringspunktet for koordinaterne i *Location* er i dette tilfælde placeret midt i vinduet og indikerer en absolut placering, i modsætning til den relative i IFC modellen. Skæringspunktet for *Facing orientation* er målt til 1 fod, fra *Location*. Efterfølgende har jeg beregnet orienteringen for at verificere den udvendige retningsvektor.

$$\text{Kvadrant: } P_{x,y} - 0.97014, 0.24253 = 2. \text{ kvadrant} \Rightarrow +180$$

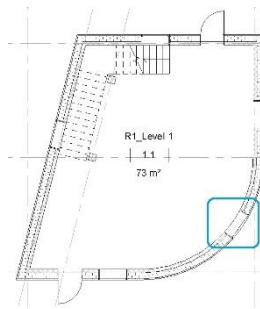
$$\text{Retning: } \tan^{-1} \left( \frac{0.24253}{-0.97014} \right) = -14.035 + 180 = 165.96^\circ$$

Tallet svarer til *Rotation* plus 90:

$$\text{Rotation} + 90 \Rightarrow 75.96 + 90 = 165.96^\circ$$

Begge ligninger resulterer i det samme facit, som stemmer overens med tallene fra IFC-modellen.

## VINDUE I RUND FACADE



Informationer omkring Location, Rotation og Orientation fremstår som for det foregående tilfælde, med undtagelse af de specifikke værdier. Ud fra disse informationer, isoleret set, kan dette vindue i princippet være placeret i lige væg, såvel som en rund. Revit Lookup beskriver vinduesobjektet som FamilyInstance Class, vist i figur 2.2.5.9.

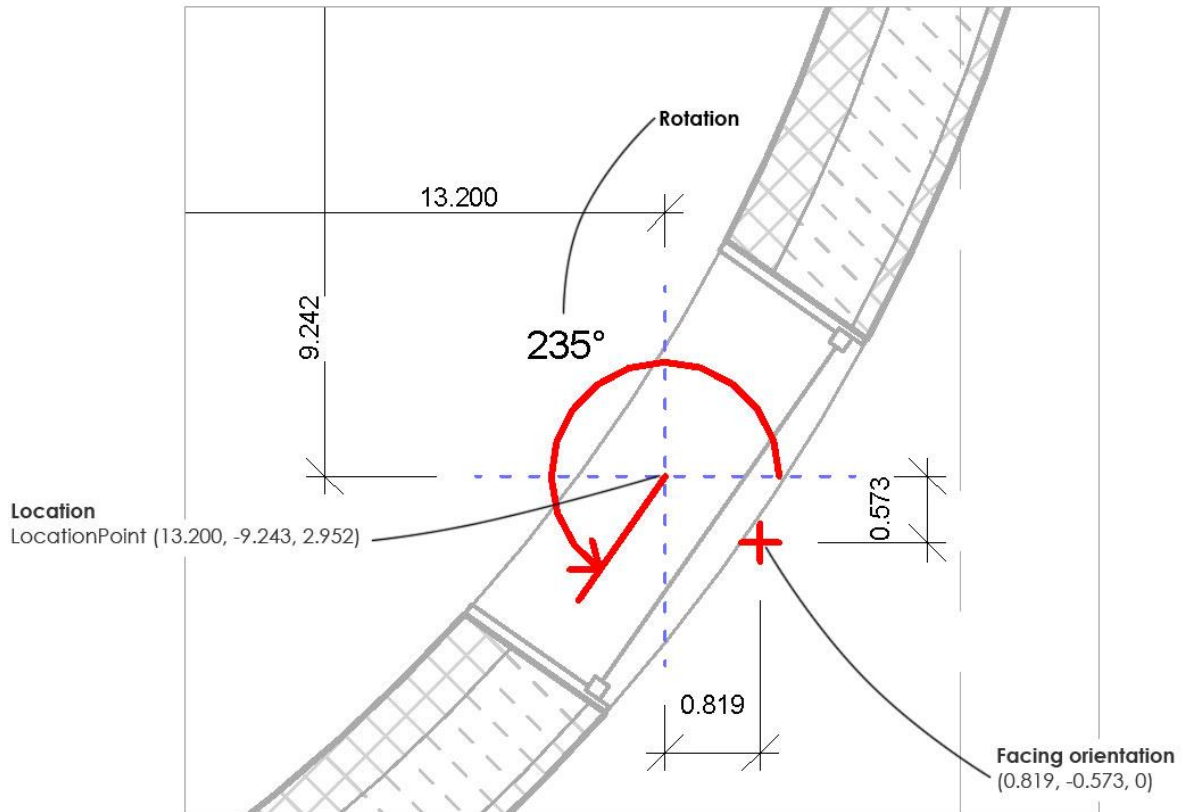
Field	Value
--- Element ---	
Name	%QQA1 - 1200x1200
ID	337156
Unique ID	fb814bca-393f-4f9f-a357-2da0f99d8385-0005...
Category	< Category >
Object type	< FamilySymbol %QQA1 - 1200x1200 3...
Level	< ElementId >
Document	< Document >
Location	< LocationPoint >
Material	< List'1 >
--- Instance ---	
Total Transform	< Transform >
Transform	< Transform >
--- FamilyInstance ---	
Host	< Wall %AD1 - Ext_450 336264 >
Symbol	< FamilySymbol %QQA1 - 1200x1200 3...
Original Geometry	< Geometry.Element >
Sub components	< List'1 >
Super component	< null >
Facing orientation	(0.819152044288947, -0.57357643635111, 0)
Hand orientation	(-0.57357643635111, -0.819152044288947, 0)
Location	< LocationPoint >
Mirrored	False
Copings	< List'1 >

Field	Value
--- APIObject ---	
Is read-only	False
--- Location ---	
--- LocationPoint ---	
Point	(13.2003609067561, -9.24299221400091, 2.95275590551183)
Rotation	234.999999999995

Figur 2.2.5.8. Snoop Current: Vindue i rund facade (FamilyInstance / LocationPoint)

Som i eksemplet med den lige facade, illustrerer den følgende figur værdier for Location og Facing orientation i sammenhæng med det specifikke vindue. De forskellige linjetyper følger det samme koncept som hidtil.



Figur 2.2.5.9. Objektgeometri og skitserede værdier ved rund facade (Location, Rotation, Facing orientation)

Skæringspunktet for *Location*-koordinaterne er ligeledes placeret midt i vinduet. En beregning af orienteringen kan se således ud.

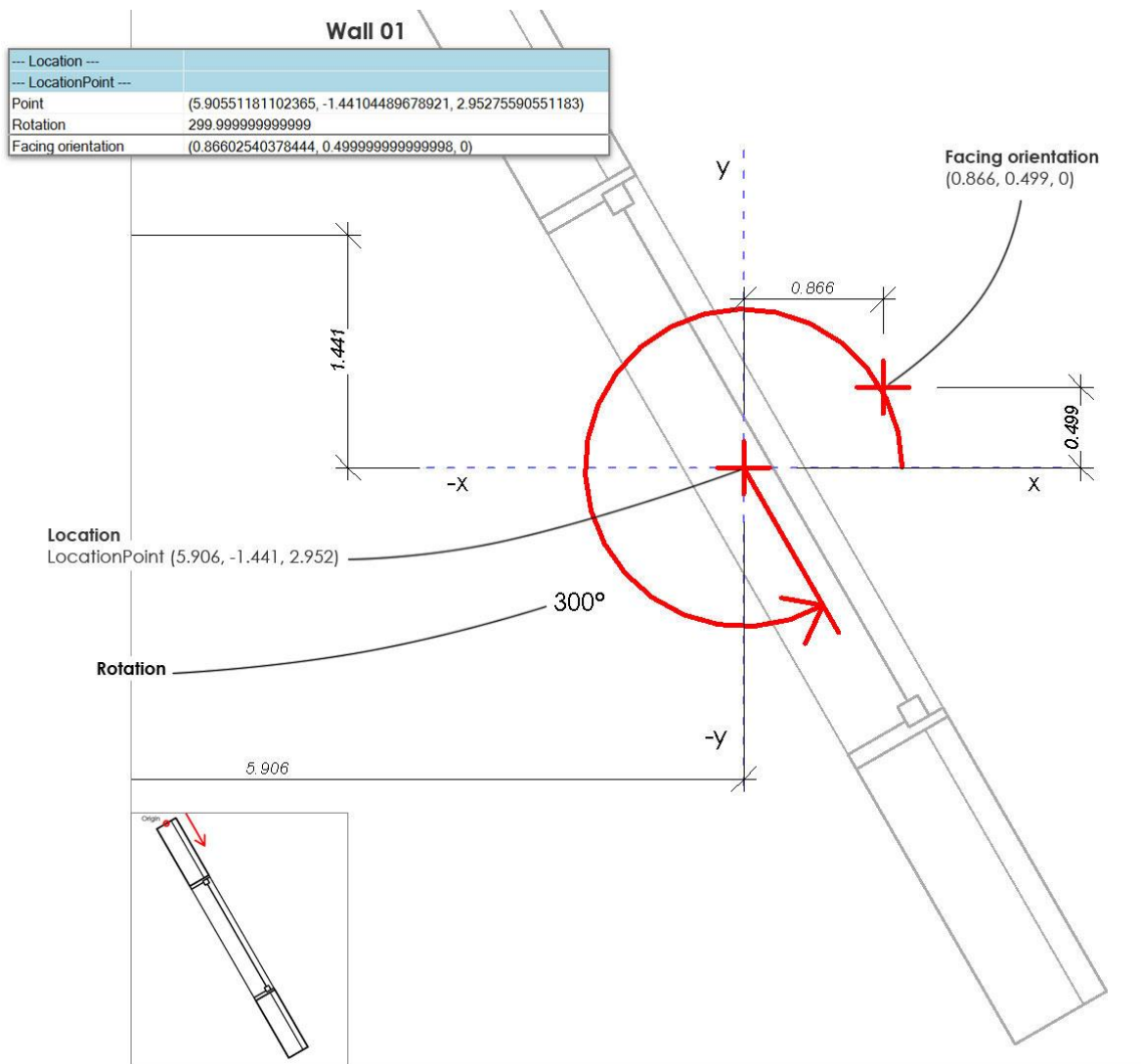
$$\text{Kvadrant: } P_{x,y} = 0.81915, -0.57357 = 4. \text{ kvadrant} \Rightarrow +360$$

$$\text{Retning: } \tan^{-1} \left( \frac{-0.57357}{0.81915} \right) = -35 + 360 = 325^\circ$$

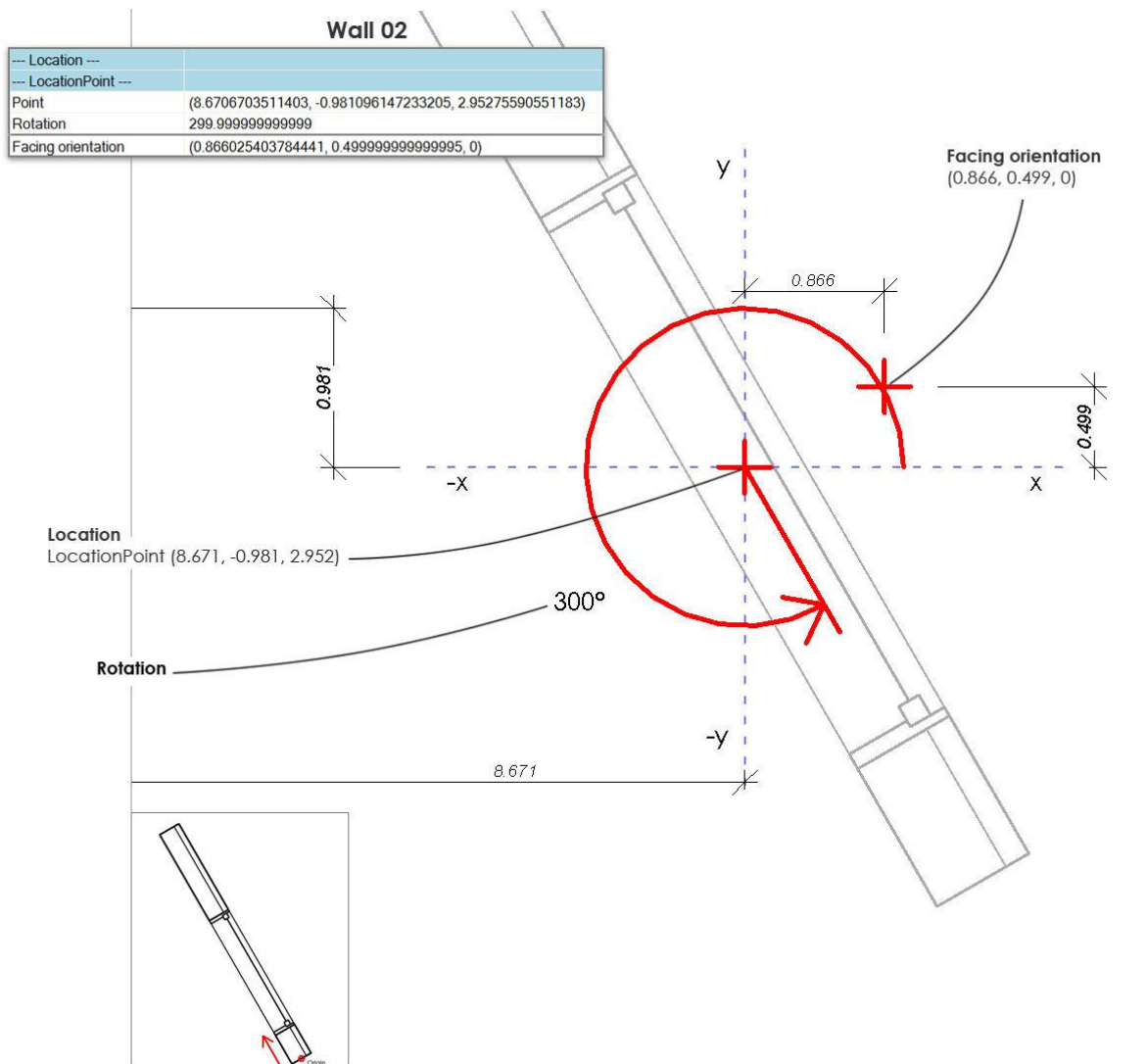
$$\text{Svarende til: } \text{Rotation} + 90 \Rightarrow 235 + 90 = 325^\circ$$

Koordinatsystemerne for de enkelte objekter er parallelle med det primære for projektet. De er således ikke roteret i forhold til relationer. Informationer om relaterede objekter (hosts etc.) er tilgængelige, men lokalisering og retning er angivet entydigt og uafhængigt af modelleringsretning.

Dette kan eksemplificeres af et forsøg med Wall01 og Wall02, som med undtagelse af location, har identiske værdier.



Figur 2.2.5.10. Wall 01: Modelleringsretning og skitserede værdier (Location, Rotation, Facing orientation)



Figur 2.2.5.11. Wall 02: Modelleringsretning og skitserede værdier (Location, Rotation, Facing orientation)

Revit Lookup viser samtidig, at væggenes offset er uden betydning i forhold til den angivne Z-værdi, der er angivet som svarende til brystningshøjden:

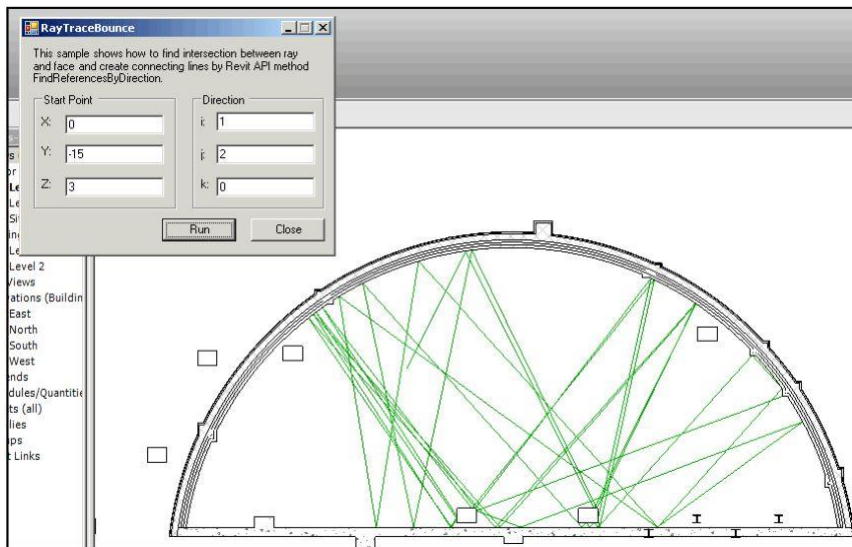
$$\underline{2.95275 \text{ ft} = 900 \text{ mm}}$$

### 2.2.5.3 POTENTIALE I API-LØSNINGER

Både litteraturen og flere SDK-samples vidner om et bredt anvendelsesområde for API'en. Det omfatter både optimering af Be10 beregning, interoperabilitet og forskellige geometriske analyser (Conover, 2010, 2011). Dette potentiale kan blandt andet relatere til følgende:

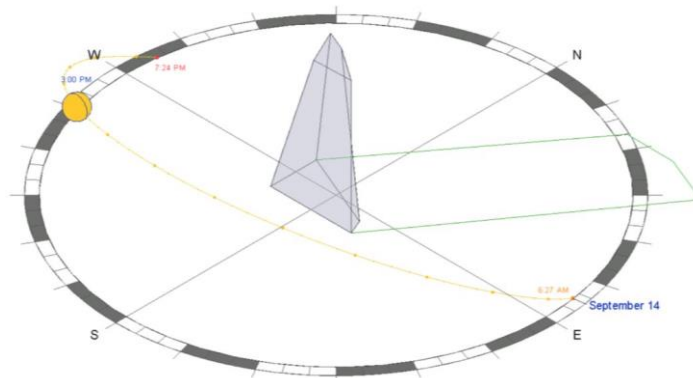
- Optimeringsperspektiv på Be10 beregningsproces
  - Skyggeberegning.
  - Dagslysberegning/faktor.
  - Definition *Space Boundary*-placering i eksempelvis vægkonstruktioner.
- Interoperabilitet i forhold til den resterende kontekst
  - Eksport af forenklet geometri til simuleringværktøjer.
  - Parameterstyring.
  - Tovejskommunikation eksport/import af værdier til/fra fx Excel (se *Appendiks F*).

Geometriske analyser, kan eksempelvis omfatte rumbelysning og reflekterende overflader:



Figur API13. Overfladeanalyse med *RaytraceBounce* (Conover, 2010, p. 18)

Et eksempel i relation til Be10 kan være skyggeberegninger ud fra analyser af geometri som nedenstående:



Figur API14. Skyggeanalyse af geometri (Conover, 2011, p. 18)

SDK-dokumentationen filen beskriver *RevitLinkInstance*, som giver adgang til informationer i linkede modeller (Autodesk Inc., 2014h). API'en er således i stand til at læse informationer fra de objekter en linket model består af. Med Revit Lookup er det muligt, at navigere frem til listen med *FamilyInstances* i den linkede model. Her er informationer præsenteret på samme vis, som de foregående eksempler med vinduer i ret og rund facade.

Funktionen er ligeledes i stand til at tilgå informationer om placering og rotation i en linket IFC-model, som indlæses betydeligt hurtigere end ved import. Informationer fremgår imidlertid under objekttypen *DirectShape* som værdier i *Transform*.

Field	Value
--- APIObject ---	
Is read-only	False
--- Transform ---	
Origin	(13.7247593708167, -7.20708038738547, 2.95275590551181)
X axis	(-0.57357643635111, -0.819152044288947, 0)
Y axis	(0.819152044288947, -0.57357643635111, 0)
Z axis	(0, 0, 1)

**Figur API15.** Revit Lookup: Transform information fra linket IFC-model læst af Revit API. De pågældende værdier tilhører vinduet i den runde facade.

Disse koordinater kan potentielt være spejlvendte som resultat af objekter der er *flipped*, hvor det i den forbindelse er muligt at sammenligne med et objekts originale XYZ-værdier (Conover, 2010). Dette kan således være nødvendigt at tage højde i forbindelse med eventuelle udtræk af data. En anden forskel, som kan være relevant at være opmærksom på, er værdierne for *Origin*, der beskriver placeringen af objektets nulpunkt. Tallene bærer præg af kombinerede IFC/Revit værdier, i den forstand, at *Origin* her refererer til et objekt-hjørne, som for IFC, mens de konkrete afstande er opgivet som absolutte, som for Revit. I dette tilfælde spiller det imidlertid ingen rolle i forhold til orienteringen, der er angivet med en værdi som hidtil og er nøjagtig ned til sidste decimal.

#### 2.2.5.4 OPSUMMERING AF METODETEST: REVIT API

Relevante informationer i forhold til fokus i analysen, er for hvert enkelt vindue tilgængelige det samme sted. FacingOrientation i API'en har specifikt til formål, at gøre orientering tilgængelig, for døre/vinduer og FamilyInstances generelt. Dette gør informationen troværdig, da softwareudbyderen har haft til hensigt, at gøre den tilgængelig. Den direkte adgang til XYZ-værdierne betyder, at tallene er genereret af systemet selv og dermed afspejler den visuelt repræsenterede information i geometrien. Et output med udgangspunkt i disse værdier beror derfor i mindre grad sammensatte beregninger af andre tilgængelige geometriske data. Det areal der i dette tilfælde er opgivet vinduerne, er derimod svær at identificere et grundlag for. Til gengæld kan information om absolutte placering være relevant i forhold til virksomhedens beskrevne eksempel med identificering af vinduer i gårdrum.

Revit Lookup viser at API'en giver adgang til en omfattende mængde af informationer i Revit og adskillige SDK-samples, suppleret af litteratur, omfatter en række eksempler på et stort potentiale i med API-løsninger, som relaterer til både Be10- parametre og perspektiver i andre dele af processen. Flere eksempler er nævneværdige, men som udgangspunkt kan funktionalitet som tovejskommunikation med Excel, definition af *Space Boundaries* og eksport af forenklet geometri til eksempelvis simuleringsværktøjer fremhæves. Tovejskommunikationen, gør det i sammenhæng med virksomhedens udviklede Be10-link, muligt at overføre data uden andre omveje end Excel.

API'en gør det muligt, at implementere løsninger direkte i Revit-brugergrænsefladen, hvilket potentielt kan gøre eksport/udtræk af informationer til en procedure bestående af ét klik med en mus. Dette potentiale omfatter ligeledes informationer i linkede IFC-modeller.

Flere eksempler på stort potentiale i form af de mange samples, artikler og mængden af synlig information i Revit Lookup. Mulighed for automatisk udtræk fra linkede IFC-, såvel som Revit-modeller. Absolut placering kan være nyttigt i forhold til eksemplet med vinduer i gårdrum.



## 2.2.6 TEST: ROCKWOOL ENERGY DESIGN

Rockwool Energy Design adskiller sig fra de øvrige metoder, som et konkret værktøj til energirammeberegning.

I testen har jeg holdt mængden af opsætning på et minimumniveau, der omfatter projektflætning og synkronisering. Nogle observationer kan derfor være forbundet med manglende konfiguration etc. BIM-link funktionen er relativt simpel at konfigurere. Jeg oprettede et projekt efter den indbyggede instruktionsvideo og flettede det sammen med testmodellen. Under denne test var jeg online med en upload-hastighed angivet til 3 Mbit/s. Sammenfletning og upload af data fra en model af denne størrelse, var uproblematisk og gennemført i løbet af få sekunder. Mine efterfølgende observationer i Rockwool Energy er beskrevet i det følgende.

### 2.2.6.1 OBSERVATIONER

#### BIM-LINK:


- Families: Bygningsdelstyper fra Revit med typenavn og tykkelse
- Kuldebroer: Bygningsdele med kuldebroer og typenavne
- Arealer: Arealtyper

Hver type kan ekskluderes fra beregningen ved at fjerne fluebenet i 'Medtag'.

I listerne for families og kuldebroer er elementerne markeret med fejl.

Et klik på markeringen ved typen for ydervægge fremkalder fejlmeddelelsen nedenfor:

Fletningsfelter opdateret fra Revit			Rockwool Energy	
Revit Architecture			Opsætning	Mål
Medtag	Family	Mål		
<input checked="" type="checkbox"/>	%AD1 - Ext_450	450 mm	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%AB1 - Ext_300	300 mm	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%BD2 - Int_150	150 mm	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%BD1 - Ext_300	300 mm	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%QQA1 - 1200x1200		<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%QQC2 - Ext_10x21		<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%QQC3 - Int_9x21		<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%BE2_590	590 mm	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%AC1_450	450 mm	<input type="checkbox"/>	
<input checked="" type="checkbox"/>	%AC2_300	300 mm	<input type="checkbox"/>	



Figur 2.2.6.1. BIM-link: Families med fejlmarkeringer

Det samme gør sig gældende for kuldebroer. I dette tilfælde var jeg i stand til at løse problemet, ved at tilknytte en tilfældig konstruktion i produktkataloget.

Under BIM-link fanen var kun arealtypen vist uden fejlmarkering.

Fletningsfelter opdateret fra Revit			
	Revit Architecture		Rockwool Energy
<b>Medtag</b>	<b>Kuldebro</b>		<b>Opsætning</b>
<input checked="" type="checkbox"/>	%AD1 - Ext_450 <> %QQA1 - 1200x1200	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	%AD1 - Ext_450 <> %QQC2 - Ext_10x21	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	%BD2 - Int_150 <> %QQC3 - Int_9x21	<input type="checkbox"/>	<input type="checkbox"/>
Fletningsfelter opdateret fra Revit			
	Revit Architecture		
<b>Medtag</b>	<b>Area Plan</b>		
<input checked="" type="checkbox"/>	Gross Building		<input checked="" type="checkbox"/>

Figur 2.2.6.2. BIM-link: Kuldebroer og Arealer

## KLIMASKÆRM

Etageantal og opvarmede etagearealer er angivet til 0,00 på trods af, at en typen bruttoareal er flettet ind fra Revit, synkroniseret og markeret 'medtag'.

**Opvarmet etageareal**

Navn

Areal  m<sup>2</sup>

Kælder

Figur 2.2.6.3. Klimaskærm: Opvarmet etageareal som manuelt input

Da jeg efterfølgende tilføjede nettoarealer og flettede projektet igen, var jeg ikke i stand til, at udelukke en sammenhæng med areal-typen. De tilføjede arealer var ikke vist i listen under BIM-link.

## KLIMASKÆRM: OPVARMET

De primære bygningsdele fra Revit er samlet i en liste med hver forekomst.

I Revit-modellen havde jeg modelleret 2 dæk. Et terrændæk og et etagedæk (figur RED4).

I Rockwool Energy er hvert dæk oplistet 2 gange (figur RED5).

<Floor Schedule>				
A	B	C	D	E
Type	Function	Area	Level	Type Comments
%AC1_450	Interior	78 m <sup>2</sup>	Level 1	Terrændæk
%AC2_300	Interior	68 m <sup>2</sup>	Level 2	Etagedæk

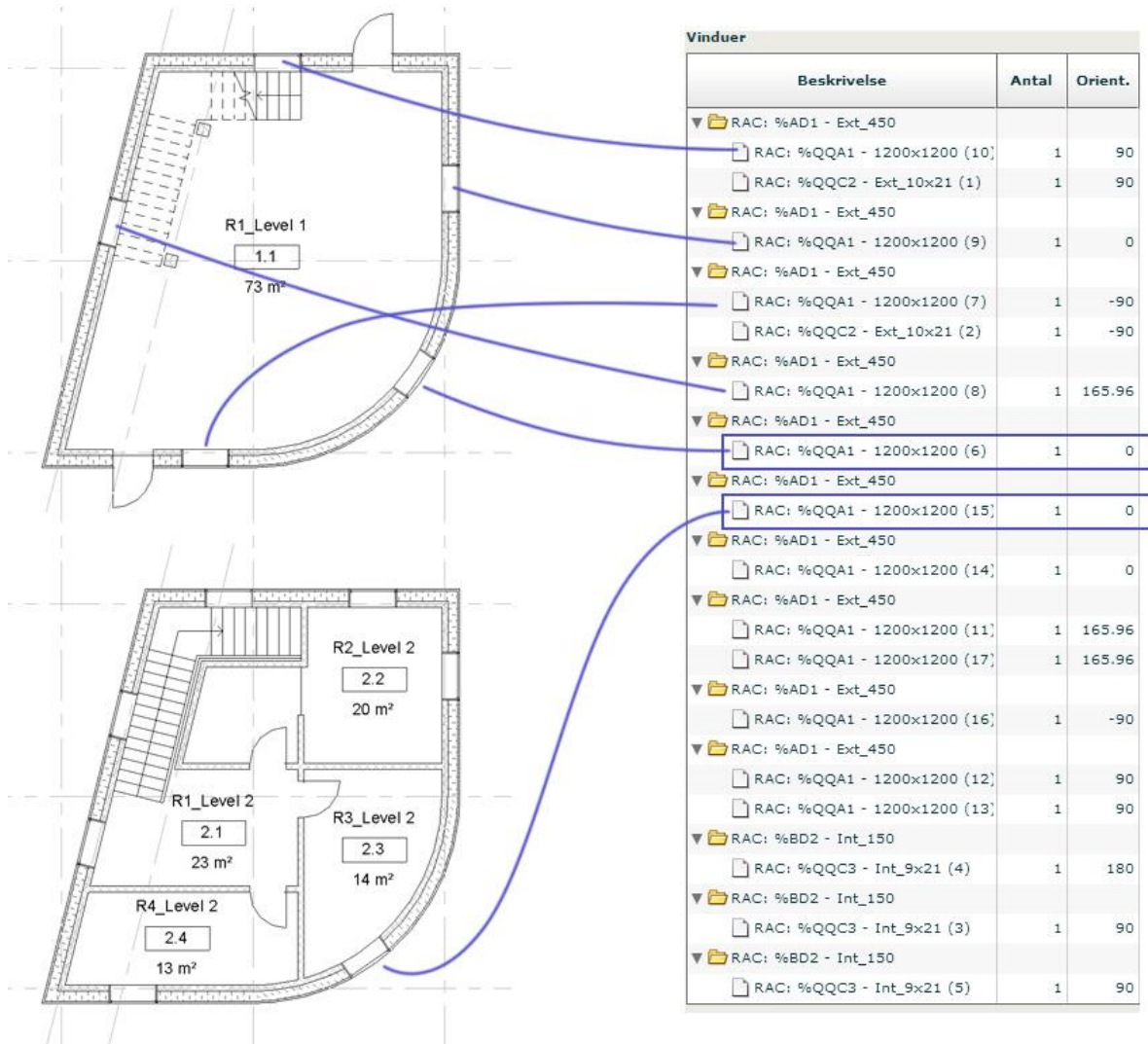
Figur 2.2.6.4. Gulvskema i Revit



Figur 2.2.6.5. Dæk i Rockwool Energy

Under listen med primære bygningsdele, er vinduer og døre vist med deres respektive *host*, og værdier svarende til et skema for vinduer i Be10. Disse omfatter orientering, som i dette tilfælde er angivet som en kombination af akse og polære grader. Forstået på den måde at værdien  $-90$  svarer til vinkelret i akse  $-Y$  og svarende til en sydvendt retning. Et eksempel på dette er vist i figur RED6, hvor et vindue med værdien er udpeget i stueplan. De repræsenterede vinduer er identificeret i Revit ved hjælp af deres værdi for 'Mark'-parameteren, som er angivet i parentes af Rockwool Energy.

De to felter jeg har fremhævet i listen repræsenterer vinduerne (6) og (15) i de runde vægge. Orientering for begge er angivet til 0, ligesom det østvendte vindue (9).



Figur 2.2.6.6. Revit-planer vs. vinduesliste i Rockwool Energy

Det vestvendte vindue er angivet til 165,96°, hvilket svarer til beregningerne med både IFC og API. Til forskel fra de to metoder, figurerer værdien her som et reelt input i energiberegningen og bør i den forstand stemme overens med SBi 213. Målt fra Nord i urets retning, resulterer orienteringen i en vinkel på 284,04°.

### 2.2.6.2 OPSUMMERING AF METODETEST: ROCKWOOL ENERGY DESIGN

Rockwool Energy Design angiver i dette tilfælde orienteringer som er uoverensstemmende med beregningsbeskrivelsen i SBI 213. For vinduerne i de runde facader er orienteringer ukorrekte. Værktøjet BIM-link funktion er ikke i stand til at håndtere opdateringer med nye supplerende arealer Revit-modellen.

Sammenkobling af bygningsdele og produktkatalog kan være en redundant proces og det er usandsynligt, at en fagperson med Be10 erfaring, kan spare tid med værktøjet. Fejlmeldinger uden henvisning til en årsag kan forsinke en beregning yderlige og en sådan risiko må anses som uacceptabel i en professionel projektorganisation med højtudannede medarbejdere og ansvar overfor samarbejdspartere.

Manglende kombination af ensartede og ensrettede bygningsdele fra Revit er en begrænsning i forhold overskuelighed i projekter af en vis størrelse. Værktøjet kan være et egnet Be10-alternativ for mindre håndværksmestre og privatpersoner, men for professionelle rådgivere, som rent faktisk anvender Revit, kvalificerer det sig ikke som en seriøs løsning. I den forstand kan BIM-link funktionen fremstå som overflødig.

De upålidelige orienteringer jeg her har demonstreret, gør i sig selv, Rockwool Energy Design uegnet i henhold til undersøgelsens problem.

### 2.2.7 DELKONKLUSION: INTEROPERABILITET OG TEST AF METODER

På baggrund af de gennemførte tests af metoder til at opnå interoperabilitet mellem Be10 og BIM med Revit som grundlag, konkluderer jeg indledningsvist at to af de testede metoder er uegnede i forhold til konteksten. Henholdsvis gbXML eksporteret fra Revit EAM og det selvstændige værktøj Rockwool Energy Design, leverer misvisende output på hver deres måde. For gbXML omfatter dette primært opdelingen af vinduer i runde facader, som al sandsynlighed er forårsaget af Revit EAM. For Rockwool Energy Design omfatter det værktøjets BIM-link funktion, der leverer ugyldige værdier i to sammenhænge: 1) Vinduer i runde vægge med orientering opgivet til 0°, og 2) værdier for alle vinduer i uoverensstemmelse med beregningsvejledningen i SBi 213.

I henhold til vinduer og fokusparametre i analysen, konkluderer jeg at IFC og Revit API, begge egner sig som metode til udtræk af informationer om orientering og hældning, samt at begge er forbundet med et behov for yderligere analyse i forhold til udtræk af pålidelige arealinformationer.

Som udgangspunkt for den efterfølgende problemløsningsfase har jeg vurderet de to metoder i en grundlæggende betragtning, som værende *metoder*, frem for *løsninger*. Med dette argument kan de således danne grundlag for en efterfølgende udvikling frem mod en potentiel løsning. Udvikling kan i den sammenhæng omfatte proces såvel som teknologi.

Der er faktorer der taler for og imod begge metoder. For IFC er eventuelle udfordringer med dataoverførsel til Excel, i forvejen løst af IFC File Analyser. For API er tovejskommunikation og entydig information om orientering, værdifulde egenskaber.

Den bedst egnede metode kan principielt være en kombineret anvendelse af begge.

Omfanget af et løsningsforslag er betinget af projektets tidshorizont. Derfor beror mit valg mellem de to metoder på en ambition om, at udvikle en deløsning som et løsningskoncept der kan udgøre et grundlag for fremadrettet udvikling. På baggrund af de forudgående analyser vurderer jeg Revit API til at leve bedst op til den ambition.

I forlængelse af dette, har jeg formuleret de afgørende faktorer for mt valg, som oplæg til en problemformulering i det følgende afsnit.

## 2.3 PROBLEMFORMULERING

---

I dette afsnit beskriver jeg de afgørende faktorer som argumentation for mit valg af Revit API til udvikling af et løsningsforslag. Dette er efterfølgende betegnet 'løsningskoncept' idet jeg udvikler en delløsning med fokus på vinduer. Beskrivelsen leder op til projektets endelige problemformulering, som er rettet mod en problemløsningsfase med udgangspunkt i det initierende problem, der er opsummeret herunder.

IFC og API forudsætter begge en målrettet udvikling for at være i stand til at udgøre løsninger på projektets problem. Mit argument er derfor at fokusere på den metode, jeg vurderer til at besidde det største potentiale. Både i forhold problemets kontekst og de relaterede processer, hvor Revit er indgår som et vigtigt element i begge. Ud fra dette er følgende taget i betragtning:

- Helhedsorienteret potentiale for teknologistøttet procesoptimering
- Ingeniørfagligt samarbejde og informationsudveksling internt i virksomheden.
- Implementering i forhold til den eksisterende ingeniørproces.

De forskellige samples i Revit SDK er samlet set et eksempel på alsidighed i funktionsmuligheder med udvikling i Revit API (Autodesk Inc., 2014h). (Conover, 2010, 2011) beskriver flere muligheder forbundet i relation til forskellige bygningsanalyser. Herunder definition af space boundaries og kalkulation reflekterende overflader og lysforhold. Disse kombineret med implementering direkte i Revit's brugergrænseflade, skaber et potentiale for API'en, som er uden for IFC-formatets funktionsområde. Potentialer relaterer til det teknologiske aspekt i den initierende problemformulering. Med ovenstående argument og projektets initierende problem som udgangspunkt, fokuserer jeg i problemløsningsfasen på procesoptimering gennem teknologi, med henblik på at udvikle et løsningskoncept i form af en Add-In til Revit 2015. Dette ender ud i den endelige problemformulering:

***Hvordan kan udvikling af API-løsninger fremme interoperabilitet mellem Revit og Be10 med henblik på teknologi-støttet procesoptimering gennem automatiseret informationsudtræk?***

Afgrænsning af problemformuleringen består i begrebet 'løsningskoncept' som dækker over en målsætning, der omfatter automatiseret udtræk af vinduesinformationer og interoperabilitet i form af output leveret til Excel.

Kapitlet er opdelt i 4 dele overordnede afsnit. Hvoraf det første gennemgår udviklingen af en teknisk løsning i form af en Add-In til Revit, programmeret i C#. Den udviklede Add-In, er i stand til at eksportere relevante vinduesinformationer fra en linket model til Excel, gennem en automatisk proces, aktiveret af brugeren med ét klik. Udviklingen følges op af en beskrivelse af afprøvning af programmet med to forskellige cases, og validering af output, som udgangspunkt for diskussion efterfulgt af projektets konklusion.



## 3.1 UDVIKLING AF EN REVIT ADD-IN

Den udviklede Add-In er ligeledes betegnet 'program' eller 'programmet' i det følgende. Afsnittet er indledt med en overordnet beskrivelse af objektorienteret programmering, efterfulgt af en formuleret målsætning og beskrivelse af overordnede trin i udviklingsprocessen, samt en gennemgang af softwaremiljø, som den grundlæggende forudsætning for udviklingen. Efterfølgende er programmets kode beskrevet i relation til specifikke informationer og anvendte metoder i programkoden.

### 3.1.1 OBJEKT ORIENTERET PROGRAMMERING (OOP)

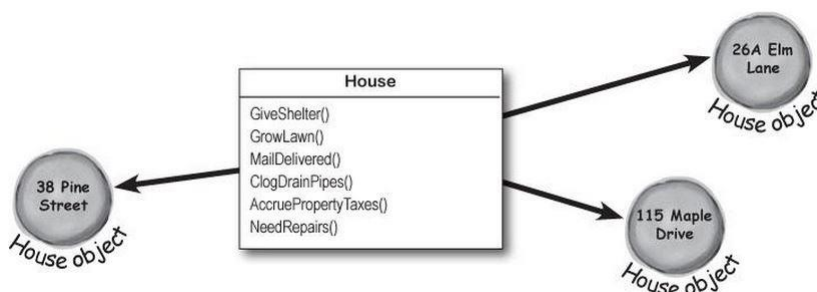
Programmering i .NET er objektorienteret, som vil sige at alting programmets kontekst er betragtet som objekter, eller dele af objekter (Greene & Stellman, 2013). Objektorienteret programmering (OOP) har flere ligheder med konceptet i BIM.



*In parametric modeling, object classes are the information structures for defining object instances. Architectural BIM design tools have object classes for Walls, Doors, Slabs, Windows, Roofs and so forth [...] The object class defines how instances of a class are structured, how they are edited and how they behave when their context changes. Another name for object class is Object Family.*

(Eastman, et al., 2011, p. 588)

I OOP benyttes *Object Classes* til at definere *Object Instances* på samme vis. Et BIM-objekt kan være associeret med en række funktioner og egenskaber, som er defineret i en *Class* eller *Family*. Det samme koncept gælder for OOP. Objekter bygger på *Classes*, der definerer *Methods* og *Properties*. *Methods* og *Properties*, er members i en class. Class er member i et *Namespace*. OOP omfatter også *Interfaces* og *Enumerations*, der kan betragtes som objekter med specifikke eller særlige egenskaber (Greene & Stellman, 2013).



Figur oop1. Class/Object princip i C# (Greene & Stellman, 2013)

Hvor BIM-objekter kan manipuleres gennem input i parametre, kan OOP-objekter manipulere data gennem variable input til en Class' metoder. Dette kaldes '*Method Overloading*', hvor variable i den forbindelse fungerer som *Arguments* (Greene & Stellman, 2013).

### 3.1.2 MÅLSÆTNING

Målet med den konkrete udvikling, er at validere Revit API, som egnet metode i forhold til opmålingsprocedurer og Be10. Denne målsætning har jeg formuleret som en række mål for konceptløsningen:

- Udtræk af relevante vinduesinformationer inkl. ovenlys
  - Areal
  - Orientering
  - Hældning
- Dataudtræk fra linket model
- Eksport til Excel
- Anvendeligt output uden behov for manuel justering
  - Bevare formatering af værdier i eksport
  - Output i overensstemmelse med SBi 213
- Simple løsning og automatisk proces

### 3.1.3 UDVIKLINGSTRIN

Jeg har udviklet programmet i trin og gradvist udbygget funktionaliteten, for afslutningsvis, at implementere funktionen til Excel eksport. Undervejs har jeg udskrevet data i Revit's brugergrænseflade, for at være i stand til at kontrollere de informationer jeg via koden, har opnået adgang til. Disse udviklingstrin omfatter overordnet:

	<b>Opnået adgang:</b>	<b>Kontroludskrift i UI:</b>
1	FamilyInstances	<i>Name og Id</i>
2	'Mark'-parameter	<i>Parameter værdi</i>
3	FacingOrientation	<i>XYZ</i>
4	Areal	<i>Areal i ft<sup>2</sup></i>
5	Linket model	<i>Filnavn</i>
6	FamilyInstances i model-link	<i>Navn, Id, 'Mark', XYZ, Areal i ft<sup>2</sup></i>
7	Excel eksport	<i>Bekræftelse af gennemført proces med opsummering af antal behandlede vinduesobjekter og varighed for kodens afvikling</i>

Tabel 3.1. Udviklingstrin i programmeringen

Hvert trin har medført individuelle processer med omregning og afrunding for at sikre brugbare og læsebare data og bevare formatering i henhold til målsætningen.

### 3.1.4 UDVIKLINGSMILJØ

I forbindelse med udviklingen har jeg benyttet et andet setup end i problemanalysens forsøg. Dette kan til en vis grad bidrage til indsigt i kompatibilitet, både for udvikling og efterfølgende løsning.

Det specifikke setup omfatter følgende:

#### OPERATIVSYSTEM

- 64-bit, Windows 8.1

#### BIM PLATFORM

- Autodesk Revit 2015, Update Release 4
- *Default installationsmappe (C:\Program Files\Autodesk\Revit 2015)*

#### SOFTWARE DEVELOPMENT KIT

- Revit 2015 SDK for UR4

#### UDVIKLERVÆRKTØJ

- Microsoft Visual Studio 2012, Update 4
- Microsoft .NET Framework 4.5

### 3.1.5 PROGRAMMERING

Afsnittet beskriver grundlæggende opsætning og programmering af den udviklede Revit Add-In. Kodeeksemplerne repræsenterer programmets faktiske kode, hvori en række metoder anvendes gentagne gange og derfor er begrænset så vidt muligt i eksemplerne.

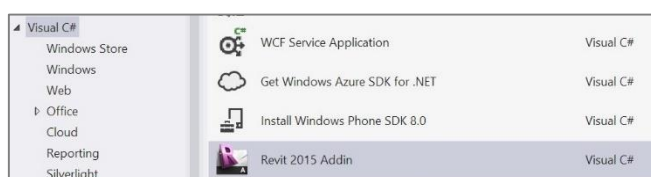
#### 3.1.5.1 INDLEDENDE OPSÆTNING

Som tidligere nævnt, understøtter Revit API flere .NET kompatible programmeringssprog. I forbindelse med den konkrete udvikling, har jeg valgt at anvende C#. Dels på baggrund af de mange SDK-samples, og dels fordi løsningen primært er inspireret og supporteret af (Tammik, 2014a) via eksempler i C#. Samtidig er introduktionsmateriale beskrevet med udgangspunkt i dette sprog (Autodesk Inc., 2014i).

Introduktionsmateriale beskriver en indledende procedure for et simpelt plug-in, men den grundlæggende kode er foruddefineret i en såkaldt 'Add-In Wizard' til Microsoft Visual Studio<sup>7</sup> (Tammik, 2014c).

Ved en default installation som er anvendt i denne sammenhæng, er installation af ovenstående *Add-In Wizard*, eneste forudsætning og kræver alene at en \*zip fil placeres i den korrekte mappe for Visual Studio Templates.

Efterfølgende er template tilgængelig for nye projekter:



Figur 3.1.5.1. Visual C# template for Revit 2015 Add-In

I forbindelse med oprettelse af et nyt projekt udfører template en række handlinger automatisk. Herunder opsætning af *Classes*, *Manifest*, *Debugging*<sup>8</sup> og *References* til API.dll og APIUI.dll-filerne i installationsmappen. Samtidig er den fornødne basiskode genereret under de oprettede *Classes*; *App* og *Command*. Dette omfatter blandt andet almindeligt anvendte Namespaces i både Windows og Revit API:

```
#region Namespaces
using System;
using System.Collections.Generic;
using System.Diagnostics;
using Autodesk.Revit.ApplicationServices;
using Autodesk.Revit.Attributes;
using Autodesk.Revit.DB;
using Autodesk.Revit.UI;
using Autodesk.Revit.UI.Selection;
#endregion
```

Figur 3.1.5.2. Visual C#: Importerede Namespaces i template-genereret kode (*Command Class*)

<sup>7</sup> Visual Studio 2012 er angivet som understøttet version i Revit 2015 API (Autodesk Inc., 2014j)

<sup>8</sup> Debug kan anvendes til fejlsøgning, hvor Visual Studio standser afviklingen og markerer det eller de steder i koden, der medfører fejl i Revit.

Derefter følger et *Namespace* svarende til det valgte projektnavn, defineret som en *subclass* til *IExternalCommand*, der implementerer programmet som en Add-In i Revit. Som hovedregel i C#, er programmets 'entry point' én metode kaldet *Main()* - Revit API benytter imidlertid *Execute()* (Greene & Stelman, 2013) (Autodesk Inc., 2014k).

```
namespace ConceptDev1
{
    [Transaction(TransactionMode.Manual)]
    public class Command : IExternalCommand
    {
        public Result Execute(
            ExternalCommandData commandData,
            ref string message,
            ElementSet elements)
        {
            UIApplication uiapp = commandData.Application;
            UIDocument uidoc = uiapp.ActiveUIDocument;
            Application app = uiapp.Application;
            Document doc = uidoc.Document;

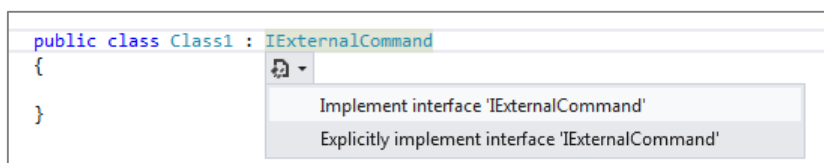
            return Result.Succeeded;
        }
    }
}
```

Figur 3.1.5.3. Visual C#: Projekt-Namespace og basiskode Add-In typen *External Tool*

*Transaction* definerer hvorledes en aktiveret Add-In figurerer i Revit's historik. Eksempelvis er det muligt at definere *Mode* til automatisk og returnere *Result.Failed*; hvorefter Revit vender tilbage til sin tidligere tilstand, inden aktivering af den pågældende Add-In. Dette sker i så fald først når koden der kommer før *return*, er afviklet.

På den måde er det muligt, at oprette midlertidige objekter gennem koden og lade Revit slette dem selv ved at annullere handlingen (Autodesk Inc., 2014j).

Visual Studio er i stand til at anvende funktionen *IntelliSense* (Microsoft, 2012a) sammen med Revit API gennem referencerne til de pågældende \*dll-filer. Funktionen kan analysere koden i realtid og præsentere en menu med kodeforslag som er mulige i sammenhæng med konteksten. Dele af koden er således mulige, at generere i større eller mindre omgang, ved at vælge forslag i *IntelliSense*-menuen:



```
public class Class1 : IExternalCommand
{
}

```

The screenshot shows a context menu with two options: "Implement interface 'IExternalCommand'" and "Explicitly implement interface 'IExternalCommand'".

Figur 3.1.5.4. IntelliSense menu med forslag til *IExternalCommand* (Autodesk Inc., 2014j)

*IExternalCommand* skal følges af metoden *Execute()*. Idet reglen er defineret i API'en, kan ovenstående eksempel således generere den tilhørende kode (Autodesk Inc., 2014j).

I .NET skal variabler deklarerer med en type, så programmet er i stand til at behandle data efter hensigten (Greene & Stellman, 2013).

```
int anyNumber = 60;
```

Ovenstående eksempel deklarerer en variabel med typen `int` (heltal) af navnet *anyNumber* og værdien *60*.

Den grundlæggende adgang til informationer i API'en er skabt af templatens. Forbindelsen er opbevaret i variablerne `app` og `doc` gennem nedenstående:

```
UIApplication uiapp = commandData.Application;
UIDocument uidoc = uiapp.ActiveUIDocument;
Application app = uiapp.Application;
Document doc = uidoc.Document;
```

Figur 3.1.5.5. Visual C#: Template-genereret kode skaber adgang til API-data opbevaret i variablerne **app** og **doc**.

Den indledende `Execute`-metode lagrer forbindelsen til Revit i variabelen `commandData`, og foretager principielt følgende:

1. `uiapp` deklarerer som `UIApplication` og forbindes med `Application`, som her er en `Property` for den Class `commandData` er tilknyttet (`ExternalCommandData`).
2. `uidoc` deklarerer som `UIDocument` og forbindes med `ActiveUIDocument`, som er en `Property` for `UIApplication`.
3. `app` deklarerer som `Application` og forbindes med `Application` under `uiapp` (som `Property` for `UIApplication`).
4. `doc` deklarerer som `Document` og forbindes med `Document` under `uidoc` (som `Property` for `UIDocument`).

De forskellige typer omfatter både `Classes` og `Properties`, og `Application` indgår flere steder og refererer noget forskelligt.

De 4 typer variablerne er deklareret med, er hver især en Class og tilhører tilsammen 3 forskellige Namespaces i Revit API, som er inkluderet med 'using' i starten af kodenn. Ved at inkludere dem på denne måde, kan de respektive Classes under hvert Namespace implementeres i koden, uden at kalde den fulde titel. Dette kan gøre ovenstående uigennemskueligt, men den generede kode resulterer i et stort omfang af relativt lettilgængelige data, gennem variablerne **app** og **doc**. De to svarer til henholdsvis: 1) Autodesk.Revit.ApplicationServices.**Application**, der repræsenterer Revit som program og giver adgang til filer, indstillinger, etc. Og 2) Autodesk.Revit.DB.**Document**, der repræsenterer et åbent projekt, og dermed en model, i Revit (Autodesk Inc., 2014m).

### 3.1.5.2 FAMILYINSTANCES

I forbindelse med problemanalysen har arbejdet med Revit Lookup på forhånd gjort det klart, at den specifikke class for de relevante vinduesinformationer er *FamilyInstance*. Analysen viste imidlertid også informationer fra den overordnede *Element* class.

Min hensigt er, at tilgå og eksportere data automatisk, med færrest mulige brugerhandlinger. Adgang til de relevante informationer forudsætter derfor at *FamilyInstance* er udpeget i koden. I kraft af, at opmåling omfatter samtlige vinduer, er jeg nødt til at iterere gennem hver forekomst.

Til dette konstruerer jeg en ny *FilteredElementCollector* og lagrer det i variabelen *wincollector*. Classen er beregnet til at filtrere og iterere gennem en samling af elementer og indeholder en særlig *Constructor* metode (Autodesk Inc., 2014m). Keyword *new*, er i dette tilfælde kaldt til denne constructor og derfor betegnet 'konstruere' et nyt objekt.

```
// construct element collection
FilteredElementCollector wincollector = new FilteredElementCollector(linkedDoc);
// specify windows as Category and FamilyInstances only
wincollector.OfCategory(BuiltInCategory.OST_Windows);
wincollector.OfClass(typeof(FamilyInstance));
```

Figur 3.1.5.6. Visual C#: *FilteredElementCollector* med specifik kategori (vinduer) og type (*FamilyInstance*)

Den efterfølgende parentes indeholder metodens *argument*, i form af en variabel deklareret som *Document* (åbent projekt). Variabelen *wincollector* indeholder i princippet alle individuelle elementer i det åbne projekt. De er derfor filtreret med metoderne *OfCategory()* og *OfClass()*. Den indbyggede kategori *OST\_Windows*, filtrerer vinduer eksplicit uafhængigt af eksempelvis sprogingstillinger (Gonçalves, 2014).

Ovenstående kodeblok samler elementer der lever op til filterens betingelser, i en liste og lagrer den i variabelen *wincollector*. *FamilyInstance*, er ikke som sådan isoleret og informationer fra relaterede classes som *Element* eller *Instance*, er forsat tilstede, men det er muligt at iterere ud fra *FamilyInstances* i en 'foreach-loop'. En 'foreach-loop', kan iterere gennem alle elementer i en liste, afvikle kode ud fra den og stopper selv ved sidste element (Greene & Stellman, 2013).

Indledningsvis opretter jeg en variabel en værdi på 0, som kan tælle objekter ved at forøge sin værdi i med én i hvert loop og udskrive det samlede antal efter sidste loop. Variabelen skal derfor deklareres uden for dens kontekst.

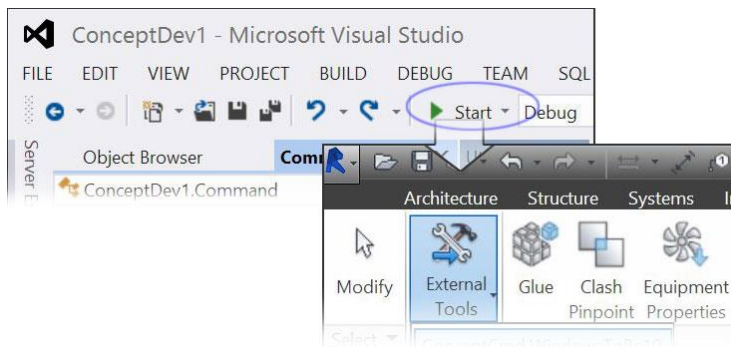
```
int linksCount = 0;
int instNum = 0;
foreach (Element elem in linkcollector)
{
    // counters
    instNum++;
}

TaskDialog.Show("Export", instNum);
```

Figur 3.1.5.7. Visual C#: Iteration med *foreach loop*

Den pågældende *foreach loop* foretager en iteration for hver *FamilyInstance* og anbringer den i variabelen *explnst*. Den afslutter med at forøge tælleren med én, i form af *instNum++*; Det samlede antal udskrives derefter med *TaskDialog.Show()*, i et vindue i brugergrænsefladen.

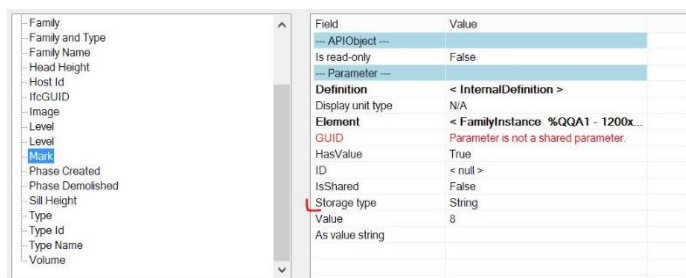
Indstillingerne i Visual Studio sørger for at Revit starter i takt med Debug og koden omdannes til et *External Tool* under *Add-Ins* fanen i brugerfladen.



Figur 3.1.5.8. Visual Studio: Start Revit med Debug

Funktionen gør det hurtigt at teste koden og kontrollere det tal programmet på nuværende tidspunkt er i stand til at udskrive, i forhold til de 12 vinduer testmodellen indeholder.

Et korrekt antal indikerer, at koden itererer efter hensigten og det således er muligt, at tilgå informationer. I den kodeblok der tilhører *foreach*, er hver *FamilyInstance* repræsenteret i variabelen *explnst*. Det er således muligt, at udføre kald til *members* af i classen. I den forbindelse er det en forudsætning, at de enkelte værdier lagres i variable, der er deklareret i overensstemmelse med *StorageType* for de enkelte parametre. Disse kan kontrolleres med Revit Lookup, hvor parameteren *Mark* eksempelvis er angivet som string (tekst):



Figur 3.1.5.9. Revit Lookup: *StorageType* for *Mark*-parameter



Name og Id er konkrete members af FamilyInstance classen og kan således kaldes direkte på expInst-variablen:

```
string elemName = expInst.Name;
ElementId elemId = expInst.Id;

Parameter paramMark = expInst.get_Parameter(
    BuiltInParameter.DOOR_NUMBER);
string markVal = paramMark.AsString();
```

Figur 3.1.5.10. Visual C#: Kald til properties og parameter i FamilyInstance (vindue)

Formålet med at hente *Mark*, er at anvende værdierne til at krydsreferere, som tidligere og på den måde identificere de enkelte vinduer i planvisninger. På trods af, at parameteren er angivet med *string* som *StorageType*, er jeg nødt til at hente den som parameter og efterfølgende omdanne den med funktionen *ToString()*. Parameteren er generel for bl.a. døre og vinduer (deraf navnet *DOOR\_NUMBER*) og således nedarvet fra *Element* classen.

*FilteredElementCollector* gør sig ligeledes gældende med et tilsvarende princip for linkede modeller. Et nyt objekt konstrueres for *LinkInstances*, som kan returnere den, eller de pågældende modeller og iterere gennem disse, som *Document* i constructor-metoden for *wincollector*. Ved at bygge kodeblokken 'ovenpå', kan *wincollector* iterere gennem objekter i hvert individuelle modellink og udtrække data fra hver enkelt for sig.

### 3.1.5.3 AREALER

Den indbyggede parameter der er tilknyttet med areal, angiver en skæv værdi.

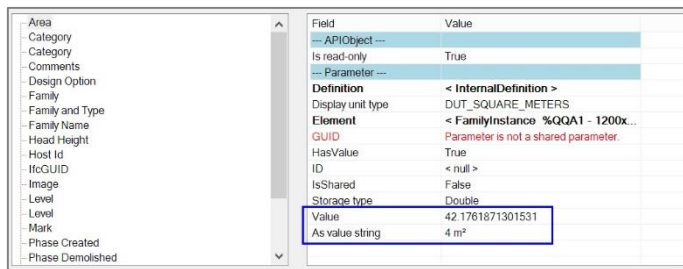
```
// get Revit's area
Parameter paramArea = expInst.get_Parameter(
    BuiltInParameter.HOST_AREA_COMPUTED);
double metricArea = Utils.areaMultiplier * paramArea.AsDouble();
```

Figur 3.1.5.11. Visual C#: Udtræk af areal-parameter

Dimensionerne på testmodellens vinduer er 1200x1200 mm og således 1.44 m<sup>2</sup>, svarende til 15.500 ft<sup>2</sup>.

Dette tal er relativt langt arealet fra API'en, der er angivet til: 42.1761871301531 ft<sup>2</sup>

Den samme værdi fremgår af Revit Lookup:



Figur 3.1.5.12. Revit Lookup: Areal parameter og værdi for vindue

En omregning til kvadratmeter resulterer i følgende:

$$1 \text{ ft}^2 = 0.09290 \text{ m}^2$$

$$42.17618 \cdot 0.09290 = \underline{\underline{3.91817 \text{ m}^2}}$$

Arealet er dermed ikke brugbart uden at kende den bagvedliggende beregning. En dør med et areal på 2.1 m<sup>2</sup> i dimensionerne 1000x2100 mm, er angivet i API'en svarende til 2.4 m<sup>2</sup>. Det tyder således ikke på, at tallene er omregnet med én bestemt faktor.

Arealet i kodens output er indledningsvis et midlertidigt resultat af *bredde x højde*.

```
FamilySymbol famSym = expInst.Symbol;
Parameter symHeight = famSym.get_Parameter(
    BuiltInParameter.FAMILY_HEIGHT_PARAM);
Parameter symWidth = famSym.get_Parameter(
    BuiltInParameter.FAMILY_WIDTH_PARAM);

double instWidth = symWidth.AsDouble();
double instHeight = symHeight.AsDouble();
double calcArea = Math.Round(instWidth * instHeight, 2);
double calcMetricArea = Utils.areaMultiplier * calcArea;
```

Figur 3.1.5.13. Midlertidigt alternativ til API-opgivet vinduesareal

Dette er uhensigtsmæssigt i forbindelse med til andet end rektangulære vinduer. En mere fleksibel metode omfatter kalkulerede overfladearealer (Conover, 2011).

### 3.1.5.4 ORIENTERING

Den første forudsætning er, at output stemmer overens med SBI 213. Orienteringen skal derfor omdannes fra de matematiske koordinater med 0° i X-aksen, til beregningsvejledningens med 0° i Nord, og svarende til Y-aksen. Samtidig er det nødvendigt at spejle beregningen, således øst, eller X-aksen svarer til 90°.

Orientering bestemmes som i problemanalysen ved beregning af den retningsvektor, API'en eksponerer i FacingOrientation.

Beregning af vektoren er foretaget med metoden *Math.Atan2* i *System.Math* (Microsoft, 2012b). For at producere et gyldigt output i henhold til SBI 213, er det nødvendigt at ombytte vektorens Y- og X-værdier i ligningen.

```
// get vector by FacingOrientation
XYZ facing = expInst.FacingOrientation;

// __Orientation__XY__
// Be10 coord system: CW with Y-axis = 0 deg
// 'typical' tan-1 is then reversed to x / y
double radiansXY = Math.Atan2(facing.X, facing.Y);
double orientDeg = radiansXY * (180 / Math.PI);
if (orientDeg < 0)
{
    orientDeg += 360;
}
```

Figur 3.1.5.14. Visual C#: Beregning af orientering med FacingOrientation

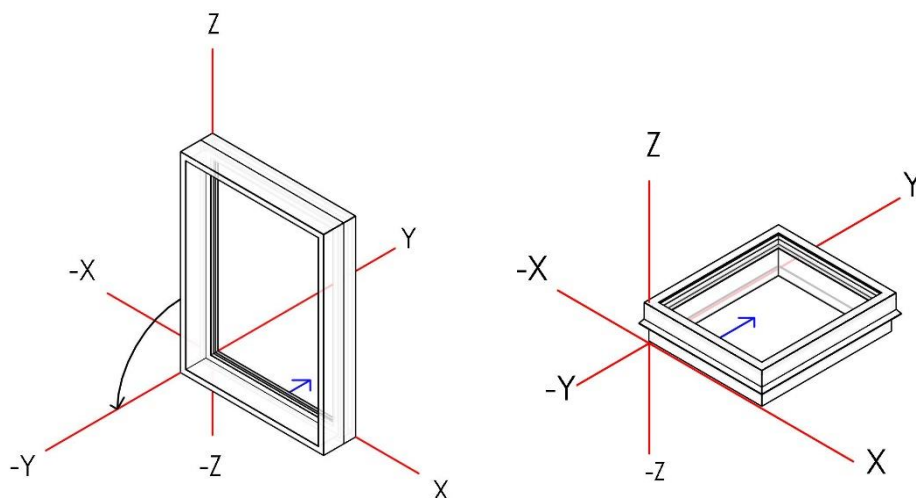
Orienteringen er suppleret med en *if-conditional*, der sikrer et korrekt output ved, at tilføje 360 i tilfælde af et negative tal i resultatet af  $radians * (180/\pi)$ .

### 3.1.5.5 HÆLDNING

Problemanalysen har ingen særlig fokus haft på hældning. Udviklingen har derfor været forbundet med supplerende analyse af forhold omkring dette. Geometri i parametriske Revit-Families, er blandt andet styret af *Reference Planes*. Disse kan hver især indgå som en specifik definition i form af center, top, bund, etc. (Autodesk Inc., 2014g).

#### ANALYSE

Vægbaseerede vinduer og dækbaseerede (som tagvinduer) er i relation til akser, forskellige i Revit. Fladen for et vægbaseret vinduer er lodret og vender med udvendig side vinkelret på Y. Dimensioner for bredde, dybde og højde fordeler sig på henholdsvis X, Y og Z. *FacingOrientation* for et tagvindue er ligeledes i Y-aksen og bredde i X. Dimensioner for *dybde* og *højde* er derimod er fordelt på akserne Z og Y.

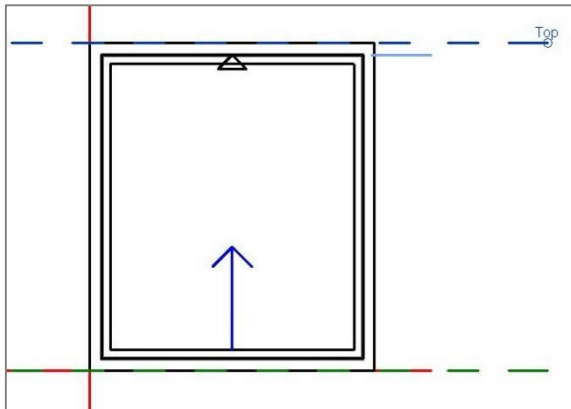


Figur 3.1.5.15a. Basiskoordinater for vægbaseret vindue. Figur 3.1.5.15b. Basiskoordinater for tagvindue. Pile angiver retningsvektor defineret af XYZ i *FacingOrientation*<sup>9</sup>

*FacingOrientation* er således forskellig fra den logiske orientering, svarende til Y og Z for henholdsvis lodret og vandret vindue i ovenstående eksempel. Dette kan besværliggøre programmeringen i forhold til en automatisk beregning der er kompatibel med begge typer.

<sup>9</sup> De angivne akser er illustreret som reference og kan afvige fra det faktiske nulpunkt for de specifikke objekter.

Ved en hældning kan det vægbaserede vindue siges, at lægges ned i en retning af -Y, roteret om X-aksen. På den måde bevares position og 2-dimensionel orientering. Tagvinduet forholder sig anderledes til hældning. Top og bund er defineret af reference planer på følgende måde:

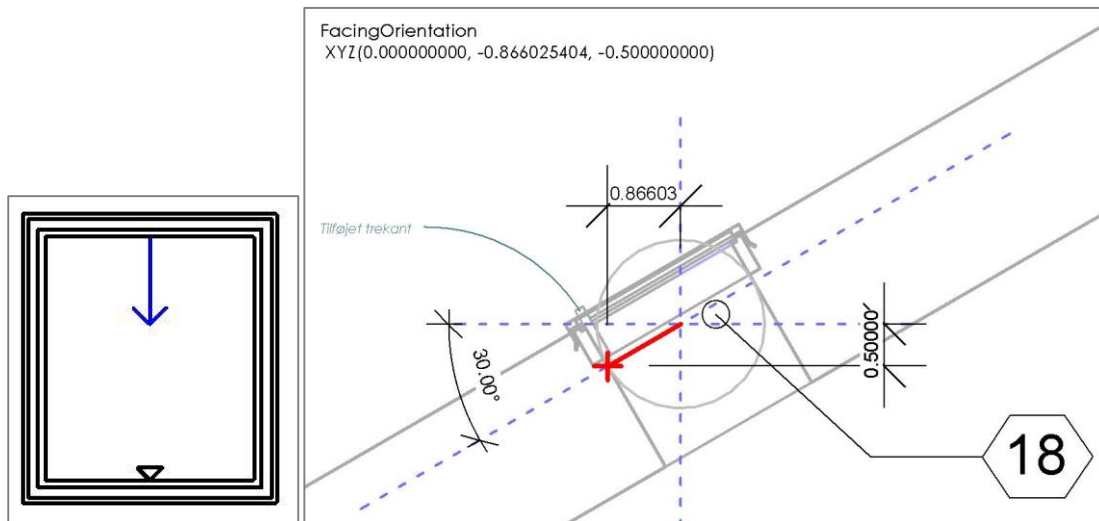


Figur 3.1.5.16. Reference planer for top og bund af et tagvindue.

Den viste Family i figur 15b har jeg suppleret med en pil i retning af Y, samt en trekant der markerer toppen af. Disse tilføjelser indgår synligt i geometrien, når den pågældende family indlæses i et projekt og kan hjælpe til at identificere objekters koordinatsystem i en kontekst.

Selvom det ene family står og den anden ligger, er basis-værdien af FacingOrientation, ved orientering mod nord, i begge tilfælde XYZ(0, 1, 0). Hvis det forestilles, at de to vinduer i figur 15 hældes med hver 45°, kan det give indtryk forskellige relative positioner i forhold til FacingOrientation. Henholdsvis på udvendig side af det vægbaserede vindue, og indvendig side af det dækbaserede tagvindue.

I forbindelse med placering af en FamilyInstance, er dette imidlertid håndteret af system-parametre og kan medføre følgende resultat:



Figur 3.1.5.17. Tagvindue i modelkontekst og retningsvektor defineret af FacingDirection.

I dette eksempel er vinduet orienteret mod syd og roteret i nedadgående retning. I henhold til førnævnte referencelinjer, vender vinduet principielt på hovedet. I forhold til modelkontekst svarer afstanden fra LocationPoint til FacingOrientation, til hældningens enhedsvektor. Dette er illustreret i figuren af en enhedscirkel med radius på 1 ft.

## PROGRAMMERET BEREGNING

Det 'nedadgående' princip for tagvinduet medfører en FacingOrientation med negativ Z-værdi, i tilfælde af vinduer med hældning.

Under forudsætning af, at tagvinduer er pålagt en minimumhældning (VELUX, 2014), er det muligt at beregne vinklen med cosinus, ved at anvende metoden *Acos* for vinduer med negativ Z. Denne tilgang kan imidlertid resultere i et misvisende output, hvor eventuelle tagvinduer uden hældning angives med 90° og kan forveksles med vægbaserede vinduer. Beregningen er i stedet betinget af vinduesobjektets *host*:

```
// Angle
double absZ = -facing.Z;
double altCosAngle = Math.Acos(absZ) * (180 / Math.PI);
string instHostCatName = expInst.Host.Category.Name;
if (instHostCatName != "Walls")
{
    altCosAngle -= 90;
}
```

Figur 3.1.5.18. Visual C#: Beregning af hældning ud fra *FacingOrientation.Z*

For at skabe et konsistent beregningsgrundlag, er en variabel deklareret som *minus Z*, til at neutralisere negative værdier. På baggrund af dette, kan den efterfølgende omregning til grader reguleres efter *host*. Dette er foretaget med en *if()* sætning, hvor 90 fratrækkes vinduer, der *ikke* er placeret i en væg. Dette kan resultere i negative vinkler, som derfor omdannes til absolutte i output med metoden *Math.Abs* og afrundes til 2 decimaler med *Math.Round*.

Med undtagelse af den forsimplede arealkalkulation, er de grundlæggende data dermed forberedt til eksport.

### 3.1.5.6 EKSPORT TIL EXCEL

Visual Studio kan indeholde referencer til en lang række af Microsofts produkter. Deriblandt findes Office, som gøres tilgængeligt ved at implementere et *Namespace* tilhørende for det specifikke program. Namespace for Excel er *Microsoft.Office.Interop.Excel*. Dette inkluderes med 'using' på samme vis som de øvrige. Den eneste undtagelse er, at det i denne forbindelse tilføjet et præfiks. Det skyldes at det indeholder flere classes med betegnelser svarende til classes i Revit API (Tammik, 2012d).

Gennem interfacet til Excel, er det muligt at oprette regneark som filer, under betegnelsen *Workbook*.

Koden initierer et nyt objekt, til at repræsentere Excel-applikationen. Objektet er lagret i variabelen *excel*. Metoder i Excel-classes kan således kaldes som eksempelvis *excel.Workbooks*.

```
X.Worksheet worksheet;

worksheet = excel.Worksheets.Add(
    Missing.Value, Missing.Value,
    Missing.Value, Missing.Value)
    as X.Worksheet;

worksheet.Name = linkName;
```

Figur 3.1.5.19. Visual C#: *Worksheet* med modelnavn og start af stopur.

I ovenstående er variabelen *worksheet* deklareret med typen *Worksheet*, der repræsenterer et enkelt ark i en fil. Hvert ark sættes til et navn svarende til den model der indgår i en given iteration.

*Missing.Value* anvendes i *arguments* som en stedfortræder for variabler der efterfølgende skriver data til det pågældende ark. Derudover er en stopurs-metode kaldt til at tage tid på eksportforløbet.

Data udskrives ved at henvise til et celleindeks i formatet [x, y] for henholdsvis række og kolonne. Relevante informationer indgår således i output ved at forbinde de enkelte variabler fra den forberedende kode, med celler i regnearket:

```
int row = 3;
int instNumLink = 0;

// iterate through collection

foreach (FamilyInstance expInst in wincollector)
{
    // range format: [RowNumber, CellNumber]

    worksheet.Cells[row, 1] = expInst.Name;
    worksheet.Cells[row, 4] = expInst.Id;

    Parameter paramMark = expInst.get_Parameter(
        BuiltInParameter.DOOR_NUMBER);
    string markVal = paramMark.AsString();
    worksheet.Cells[row, 5] = markVal;

    // ... splitline ...
```

Figur 3.1.5.20. Visual C#: Navn, Id og Mark, skrives til Excel-kolonne 1, 4 og 5

Et givent række-nummer udfyldes af variabelen *row*, der repræsenterer et nummer for den enkelte iteration gennem samlingen af *FamilyInstances*. Variablen sættes indledningsvis til 3, og forøges i slutningen af hvert loop, på samme vis som tælleren oprettet tidligere.

De resterende variabler er tilsvarende forbundet til output i celler, hvoraf den sidste er XYZ-værdier for FacingOrientation, således output kan kontrolleres ved en manuel beregning.

*Foreach*-blokken for vinduer afsluttes med at forøge lægge 1 til *tæller*-variabler med ++.

Kodeblokken for den *Foreach*, der itererer gennem modellinks, afsluttes efterfølgende med at skrive filnavn for en given model og opsummere antallet af vinduer eksporteret fra denne.

```

        // XYZ as string
        worksheet.Cells[row, 15] = facing.ToString();

        // counters
        instNum++;
        row++;
        instNumLink++;
    }

    // one more counter
    linksCount++;

    worksheet.Cells[1, 1] = instNumLink;
    worksheet.Cells[1, 2] = "instances from link: " + linkedDoc.Title;
}

sw.Stop();

string exportMsg = "Total Window Instances exported: " + instNum +
    "\n From " + linksCount + " Linked Documents" +
    " in " + sw.Elapsed.TotalSeconds + " seconds";

TaskDialog.Show("Export", exportMsg);

// the end
return Result.Succeeded;
}
}
}

```

Figur 3.1.5.21. Visual C#: Afrundede loops, opsummeret output og afsluttende bekræftelse i Revit UI.

Programmet er afsluttet med at stoppe uret og opsummere eksporteret data i en dialogboks i brugergrænsefladen.

Med undtagelse af to metoder, defineret i en separat Class, repræsenterer foregående beskrivelse den konkrete kode i programmet, som efter følgende af testet på to cases.

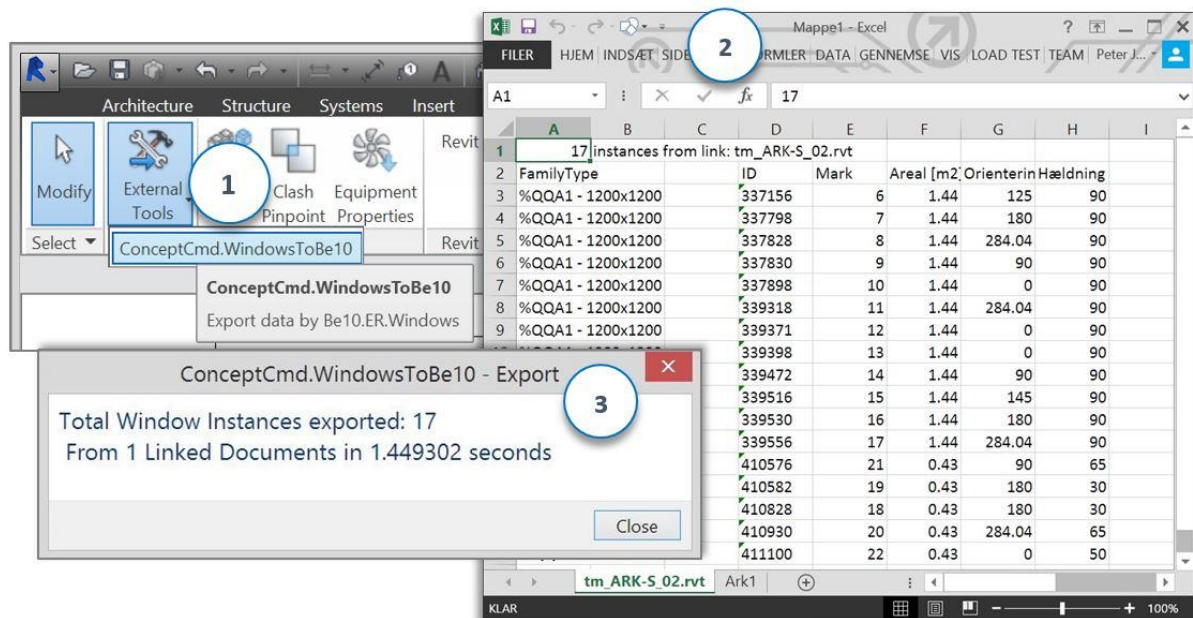


## 3.2 AFPRØVNING OG CASES

Programmets funktionalitet og output er afprøvet på to forskellige cases, som omfatter en tilpasset version af testmodellen i Case 2 og en konkret arkitektmodel, Case 3. Brugeraktivering af programmet er beskrevet indledningsvist, hvorefter cases, funktion og validering af output følger. Proceduren for afprøvning og validering har til formål at skabe grundlag for en vurdering af output i forhold til gyldighed og pålidelighed, som led af en samlet vurdering af den udviklede Add-In, som et konkret og relevant løsningskoncept og grundlag for videre udvikling. Resultater er beskrevet, som afsluttende oplæg til den efterfølgende diskussion.

### 3.2.1 ADD-IN OG FUNKTIONALITET

Programmet aktiveres i *Ribbon > Add-Ins > External Tools(1)*. Efterfølgende starter Excel automatisk(2), før Add-In-programmet afslutter med en dialogboks i Revit, hvori eksporteret data er opsummeret med en angivelse af processens varighed.



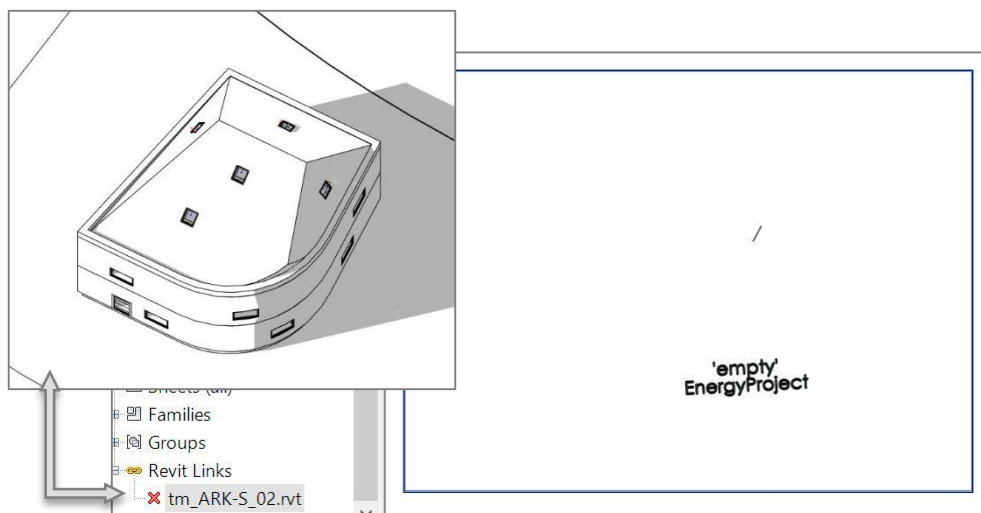
Figur 3.2.1.1. Aktivering af Add-In under External Tools, output til Excel og afsluttende dialogboks i Revit UI

Overførslen af data fra Revit til Excel, er i dette tilfælde gennemført med en varighed på 1½ sekund for de 17 vinduer. Resultatet i Excel er nedenstående regneark, hvor navnet er tilsvarende den linkede testmodel.

### 3.2.2 TESTMODEL CASE 2

For at være i stand til at efterprøve programmets formel til beregning af vindueshældning, har jeg foretaget en række mindre justeringer i testmodellen.

Justeringer omfatter specifikt en ændring af taget fra 3° ensidigt fald, til flersidet hældning i 3 forskellige vinkler på henholdsvis 30, 50 og 65°. Derudover er 5 tagvinduer placeret i de 4 tagflader, som illustreret i nedenstående figur.



Figur 3.2.2. Isometrisk billede af tilpasset testmodel Case2

Den tilpassede model er efterfølgende gemt med navnet '*tm\_ARK-S\_02.rvt*' og indlæst som link i et tomt Revit-projekt, der repræsenterer en template til formålet.

Testmodellen er imidlertid opbygget i en kontrolleret proces med det specifikke formål, at være genstand for undersøgelse. Den er således ikke repræsentativ for uforudseelige forhold, som resultat af eksempelvis projektskala, flersidig påvirkning fra samtidige/samarbejdende brugere og menneskelig handling.

17 instances from link: tm_ARK-S_02.rvt					
FamilyType	ID	Mark	Areal	Orientering	Hældning
%QQA1 - 1200x1200	337156	6	1.44	125	90
%QQA1 - 1200x1200	337798	7	1.44	180	90
%QQA1 - 1200x1200	337828	8	1.44	284.04	90
%QQA1 - 1200x1200	337830	9	1.44	90	90
%QQA1 - 1200x1200	337898	10	1.44	0	90
%QQA1 - 1200x1200	339318	11	1.44	284.04	90
%QQA1 - 1200x1200	339371	12	1.44	0	90
%QQA1 - 1200x1200	339398	13	1.44	0	90
%QQA1 - 1200x1200	339472	14	1.44	90	90
%QQA1 - 1200x1200	339516	15	1.44	145	90
%QQA1 - 1200x1200	339530	16	1.44	180	90
%QQA1 - 1200x1200	339556	17	1.44	284.04	90
%QQA3 - 0610x0700	410576	21	0.43	90	65
%QQA3 - 0610x0700	410582	19	0.43	180	30
%QQA3 - 0610x0700	410828	18	0.43	180	30
%QQA3 - 0610x0700	410930	20	0.43	284.04	65
%QQA3 - 0610x0700	411100	22	0.43	0	50

Tabel 3.2.1. Excel: Vinduesinformation oplistet i ark opkaldt efter model (her *tm\_ARK-S\_02.rvt*)

Input i Excel er formateret i henhold til datatype i de deklarerede variabler for hver enkelt parameter. Dermed er det muligt at anvende værdier som areal, orientering og hældning i en videre beregning, uden en forudgående manuel tilpasning af celleformateringer.

### 3.2.3 KONTROL AF OUTPUT

De enkelte vinduer er som i foregående analyser, identificeret ved hjælp af 'Mark'-værdier som krydsreference. Efterfølgende er Be10-relaterede værdier kontrolleret ved manuel opmåling, hvor det er relevant. I dette tilfælde orienteringer generelt, samt hældninger på tagvinduer.

Opmåling er i henhold til SBi 213:

Orientering: 0° = Nord, 90° = Øst, 180° = Syd og 270° = Vest

Hældning: 0° = Vandret, 90° = Lodret

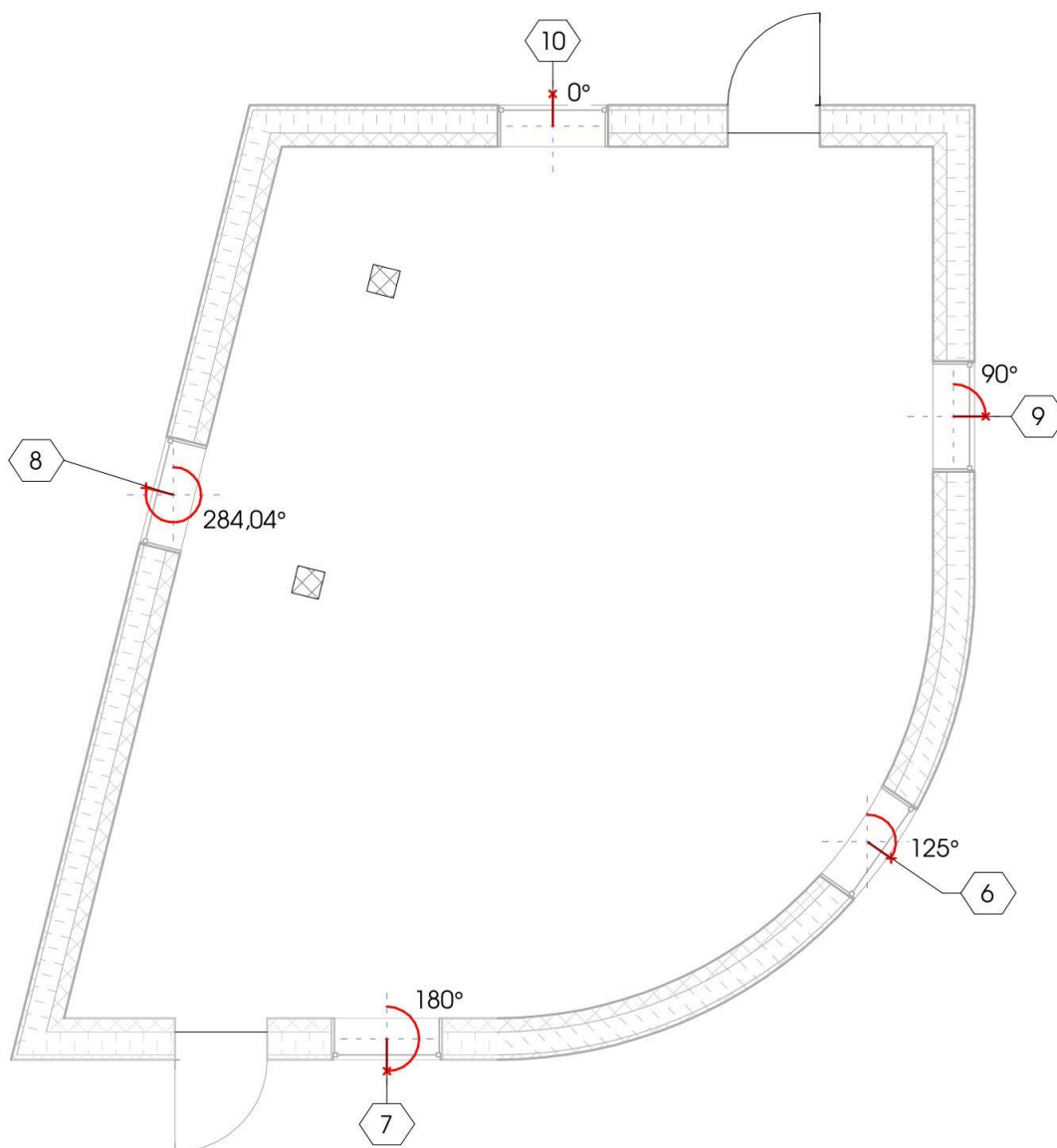
En række målte orienteringer og hældninger er i det følgende illustreret i sammenhæng med værdier Excel.

Appendiks G beskriver opmåling og Excel-ark i samlet omfang.

ORIENTERING: STUEPLAN

Mark	Areal	Orientering	Hældning
6	1.44	125	90
7	1.44	180	90
8	1.44	284.04	90
9	1.44	90	90
10	1.44	0	90

Tabel 3.2.3.1. Vinduesdata i Excel-ark, stueplan

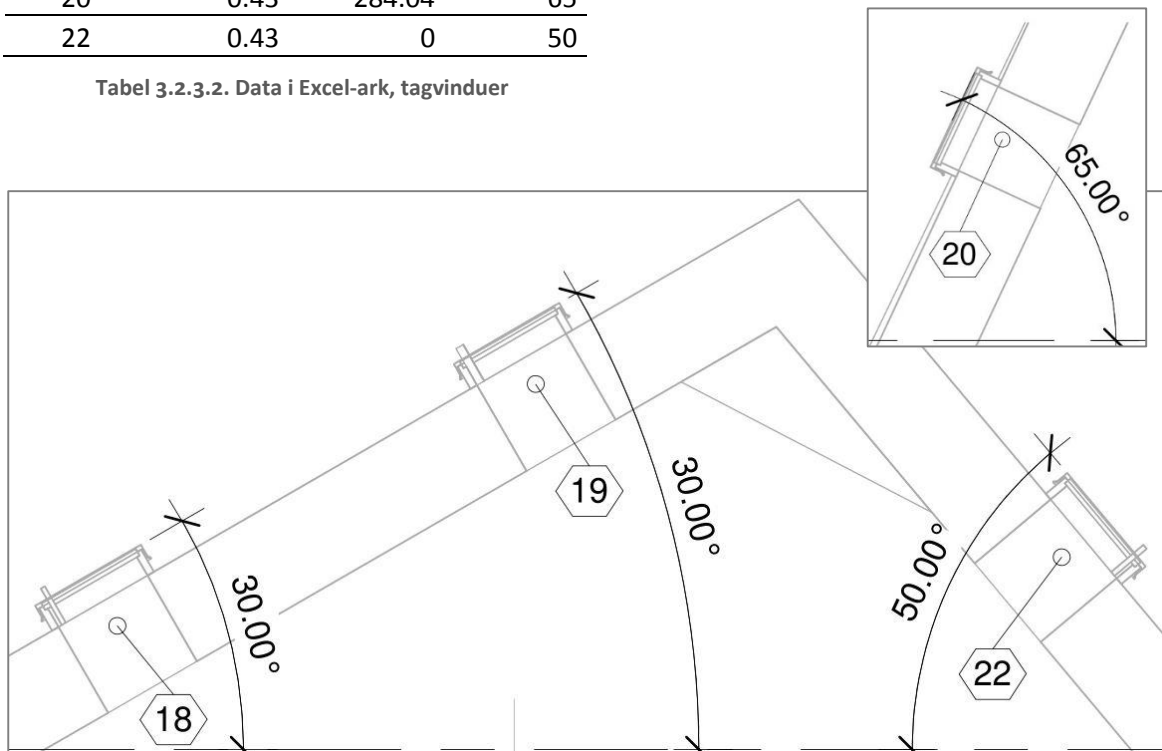


Figur 3.2.3.1. Vinduer, 'Mark'-tags og opmålte orienteringer i stueplan

## HÆLDNING

Mark	Areal	Orientering	Hældning
21	0.43	90	65
19	0.43	180	30
18	0.43	180	30
20	0.43	284.04	65
22	0.43	0	50

Tabel 3.2.3.2. Data i Excel-ark, tagvinduer

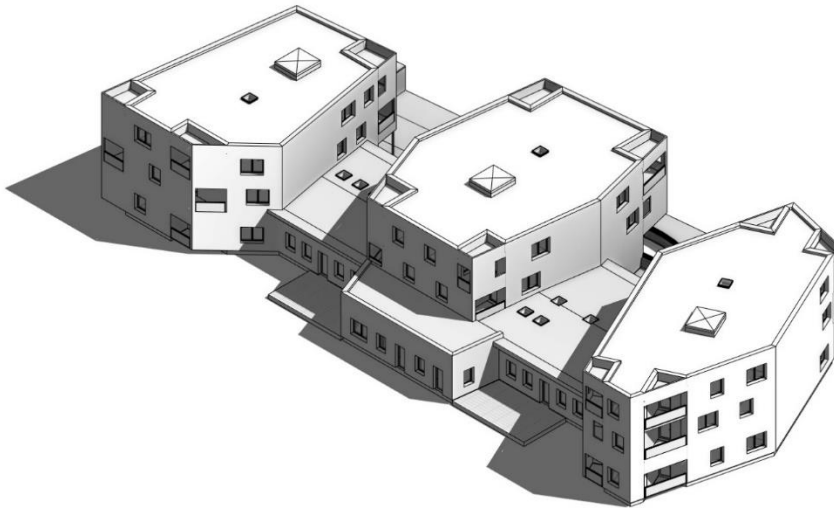


Figur 3.2.3.2. Tagvinduer, 'Mark'-tags og opmålte hældninger i snit

Som illustreret i figurerne ovenfor, er af de målte orienteringer og hældninger overensstemmende med output fra Revit/input i Excel, gennem den udviklede Add-In.

### 3.2.4 CASE 3

Jeg har haft mulighed for at gennemføre en test med faktisk arkitektmodel, fra et konkret byggeprojekt. Den er således et resultat af løbende udvikling i en samarbejdsproces. Modellen er på niveau med det afleverede projektforslag i forhold detaljeringsgrad og informationsmængde. Bygningen rummer i alt 107 vinduer i flere forskellige facadevinkler.



Figur 3.2.4.1. Case 3: Isometrisk billede testet arkitektmodel (CUBO Arkitekter, 2013).

Arkitektmodellen er linket ind i den samme model, som testmodellen, hvorefter Add-In-programmet er aktiveret på ny. I dette tilfælde eksporteres vinduesdata fra begge de linkede modeller til Excel:

	E	F	G	H	I	J	K	L	M
1									
2	Mark	Areal [m2]	Orienterin	Hældning		X:	Y:	Z:	
3	9	1.74	354.47	90		-0.09637	0.995346	0	
4	10	2.62	277.35	90		-0.99178	0.12793	0	
5	12	2.62	136.75	90		0.685183	-0.72837	0	
6	14	2.62	180	90		-3.2E-15	-1	0	
7	16	1.44	180	90		-3.2E-15	-1	0	
8	17	1.44	180	90		-3.2E-15	-1	0	
9	18	1.74	103.47	90		0.972492	-0.23294	0	
10	19	2.62	103.47	90		0.972492	-0.23294	0	
11	20	2.62	64.87	90		0.905347	0.424674	0	
12	21	2.62	283.47	90		-0.97249	0.232936	0	
13	22	1.44	96.91	90		0.992736	-0.12031	0	
14	24	1.44	180	90		-3.2E-15	-1	0	
15	25	1.44	180	90		-3.2E-15	-1	0	
16	26	1.44	180	90		-3.2E-15	-1	0	
17	27	1.44	180	90		-3.2E-15	-1	0	
18	29	0.85	244.87	90		-0.90535	-0.42467	0	
19	30	0.85	244.87	90		-0.90535	-0.42467	0	
20	37	2.62	0	90		3.3E-15	1	0	
21	38	0.85	13.47	90		0.232936	0.972492	0	

ConceptCmd.WindowsToBe10 - Export

Total Window Instances exported: 124  
From 2 Linked Documents in 2.5210602 seconds

Close

Figur 3.2.4.2. Case 3: Output med 107 vinduer til Excel (Ovalcirkel markerer selvstændigt ark for testmodellen)

De gråtonede celler i ovenstående figur, er efterfølgende valideret, som stikprøver ved opmåling i arkitektmodellen. Tagvinduer i denne model har ingen hældning og er angivet til 0° i Excel-arket og antyder dermed at koden til beregning af hældning leverer output efter hensigten.

### 3.3 RESULTATER

---

Afprøvning af den udviklede Revit Add-In med de beskrevne cases har resulteret i entydige data, som er kontrolleret og valideret gennem manuel opmåling i de konkrete modeller.

De kontrollerede data for testmodellen omfatter alle relevante værdier i forhold til programkoden.

Samtlige opmålinger har svaret til programmets output ved aflæsning i Excel.

De kontrollerede værdier i programmets output fra case 3 (arkitektmodel) har ligeledes været overensstemmende. Den 3. case viste desuden en tendens for vandretliggende tagvinduer, der antyder at programkoden håndterer hældningsberegning og skildring af objekttyper efter hensigten.

Idet jeg ikke har været i stand til at konstatere fejlagtige data, betragter jeg output fra den udviklede Add-In som valide i forhold til opmåling af vinduesorientering og hældning.

## 3.4 DISKUSSION

---

Resultater af afprøvning med de to cases har været korrekte i forhold til opmålte orienteringer og hældninger. Det er imidlertid ikke dermed sagt, at programmet er kvalificeret til anvendelse i den nuværende form.

Begrebet 'koncept' er mere end et supplement i en betegnelse af programmet, det har ligeledes været toneangivende for den udviklingsproces der ligger til grund for den Add-In version der er afprøvet. Forstået på den måde, at en kombination af den snævre tidsplan, og afgrænsede fokus har medført, den kortest mulige vej fra målsætning til resultat, som udelukkende har været mulig fordi jeg har besiddet rollerne som både udvikler og tester, og dermed har haft indsigt i forudsætningerne for en succesfuldt afviklet kode.

Et ord som 'exceptions', er relateret til fejlhåndtering af uforudseelige fejl i afvikling af en kode og noget de fleste programmører med al sandsynlighed er bekendt med. Den udviklede add-in tager generelt ikke højde for fejl. Koden vil eksempelvis resultere i et tomt regneark, hvis et projekt ikke indeholder linkede modeller, og konkrete fejle, hvis en linket model ikke er indlæst. Dette skyldes projektets varighed og er ikke som sådan betragtet som et problem.

Det er derimod de arealer. Dels arealer i API'en, til dato er uopklarede. Og dels den forenkede beregning i koden. Den simple beregning er i stand til at levere et korrekt output under de rigtige forudsætninger, som begrænser sig til rektangulære vinduer. Koden vil givet afvikles for vinduer med andre former uden at fejlmeldinger, mens outputtet i så fald er misvisende. I den forbindelse kan en fejlmelding potentielt være at foretrække, således en forkert beregning i output forhindres i indgå i anden sammenhæng.

Programmets brugbarhed kan ligeledes være relateret til andre mere kontekstbaserede problematikker, i forhold til eksempelvis tagvinduer og hældning. Beregningen i koden tager højde for den konceptuelle forskel på væg- og dækbaserede vinduer, men i tilfælde af sidstnævnte uden hældning, er har det i princippet ingen orientering. I et tværfagligt projektteam kan det i så fald være hensigtsmæssigt, at gøre en ansvarlig aktør opmærksom på et eventuelt problem. Eksempelvis i form af minimumhældninger der ikke overholdes for tagvinduer. Omvendt kan det være ubetydeligt i tidlige designfaser, hvor beregninger kan være tiltænkt som 'benchmarks' for vurdering af konsekvenser af designændringer og energioptimeret design.

Overvejelser som disse kan principielt være forbundet med automatisering i et generelt perspektiv, der ligger uden for fokus i denne undersøgelse, men en sådan problematik være relevant for videre programmet som grundlag for videre udvikling.

En problematik som arealet i API'en kan være anderledes presserende i relation til dette projekt, idet en forespørgsel til softwareudbyderens udvikler-support, om det systemgenererede areal, resulterer i en henvisning til polygonberegning som løsning (Tammik, 2013e). En løsning er således tilgængelig, men hvis et areal opgives forkert af systemet, kan det i princippet gælde andre værdier.

Andre bearbejdede parametre, er derimod entydige og uanset, erfaring og indsigt, kan manuel opmåling umuligt konkurrere med data eksporteret for 124 vinduer i løbet af 2½ sekunder. Hvor af den parameter med størst indflydelse på aktørers samarbejde og koordineringsbehov, er valideret ved samtlige kontroller udført i forbindelse med afprøvning. Og selvom programmet endnu mangler at kunne korrigere for et roteret 'sandt nord', kan funktionen være af betydelig værdi som redskab til at mindske redundans i processen ved at reducere behovet for at anvende lokale model-kopier.



# KONKLUSION

---

Projektet beskrevet i nærværende rapport, behandler én tilgang til de i starten opridsede problemstillinger. Denne tilgang har ført mig igennem en proces med varierende fokus i forbindelse med forskelligartede analyser.

Min problematisering af emnet ved projektets begyndelse har medført en initierende problemformulering, som til stadighed har været gældende ved problemformulering forud for løsningsfasen og frem til projektets afslutning.

***Hvordan skabes en kobling mellem Revit og Be10, som kan automatisere manuelle opmålinger og samtidigt reducere behovet for supplerende information i lokale modelkopier?***

Særligt det beskrevne koncept for anvendelse af lokale kopier, har været mulig at identificere. Både i forhold til case-virksomhedens udviklingsprojekt og procesanalyser. De analyserede processer, bærer præg af at indsatsen for at fremme anvendelse af teknologi, på sin vis bremses af restriktioner i selv samme teknolog. Disse restriktioner til en vis grad logiske i sammenhæng med rådgiverfaglige ansvarsområder og efterlader et mismatch mellem objektbaserede principper og ansvar for informationer, som et vedvarende problem. I forbindelse med procesanalysen har jeg været i stand til at identificere vinduesorientering som en årsag til komplekse procedurer for opsætning af den teknologi, der har til formål at fremme processen.

Gennem min tilgang til problemet, som fokuserer på automatiseret udtræk af tilgængelige information, har jeg gennemført test af forskellige metoder med henblik på interoperabilitet og BIM, i sammenhæng med CAD og Be10. Procesanalysen har argumenteret kraftigt for at fokusere på vinduesparametre i forbindelse med test af metoder. Tests og tilhørende analyser konkluderer for det første, at kilder til problemer kan omfatte faktorer ud over et givent format. Problemet er specifikt forbundet med gbXML i de testede metoder, men understreger betydningen af et konkret CAD-værktøj i konteksten. I forbindelse med de gennemførte test har jeg desuden konkluderet at Rockwool Energi Design er forbundet med visse uoverensstemmelser imellem dets BIM-link og Be10. Testfasens konkluderer afslutningsvis at både IFC og API er i stand til at levere de relevante informationer.

Et efterfølgende prioriteret valg af metode til videre anvendelse i problemløsningsfasen er truffet på baggrund af den grundlæggende betragtning af metoderne som ikke værende løsninger. På den baggrund har problemets kontekst være afgørende i form af Be10 betragtninger af bygninger og Revit som betydende elementer i en vurdering fremadrettet potentiale.

I problemløsningsfasen har jeg arbejdet med udvikling af et teknologisk værktøj til at fremme procesoptimering gennem automatisering og i den forbindelse identificeret både fordele og u hensigtsmæssige forhold i udviklingen med Revit API, som har resulteret i et løsningsforslag under betegnelsen løsningskoncept. Betegnelsen dækker over programmets ufuldendte funktionalitet, som er uddybet i diskussionen, som ligeledes behandler resultater af programmets afprøvning to

cases. I den forbindelse diskuterer jeg samtidig programmet i relation overvejelser og funktionalitet, med fordele og ulemper af forskellig grad.

Den udviklede Add-In kan diskuteres i flere sammenhænge. Men betydningen af en enkelt parameter for store dele af en proces og projektorganisation kan være svær at formulere.

Status for det udviklede program taget i betragtning, er jeg af den klare overbevisning at hvert af de diskutererede fordele og ulemper er sandsynlige at løse.

For at svare på projektets problemformulering må jeg forholde mig det projektforløb jeg har gennemgået, hvor jeg undervejs har demonstreret en funktionalitet i programmet, som har leveret entydige og valide resultater for den Be10-parameter, der udgør det absolut største problem i processen.

***Hvordan kan udvikling af API-løsninger fremme interoperabilitet mellem Revit og Be10 med henblik på teknologi-støttet procesoptimering gennem automatiseret informationsudtræk?***

Som svar på projektets problemformulering, argumenterer jeg for at det udviklede program på trods af en beskedent funktionalitet, besidder en evne til at udtrække informationer om orienteringer effektivt. I kraft af parameterens betydning for projektorganisation og proces konkluderer der således, at den udviklede API-løsning beviselige besidder et potentiale, der kvalificerer sig som udgangspunkt for videre udvikling.

Som supplement til dette argument skal det nævnes at videre udvikling i den forbindelse ligeledes omfatter programkoden i den udviklede Revit Add-In.

## Figurer

Figur 1.3.1. MOE Basisorganisation (MOE, 2014b).....	5
Figur 1.3.2. MOE Projektorganisation (MOE, 2014b) .....	5
Figur 1.4.1. Konceptuel procesdiagram: Dokumentbaseret udveksling af modeller i en principiel designproces for arkitekt og ingeniør .....	9
Figur 2.1.1. Leavitt model [efter (Leavitt & Bahrami, 1988)].....	12
Figur 2.1.2. Eksempel på flowmodel (Beyer & Holtzblatt, 1998, p. 93).....	13
Figur 2.1.3.1. Workflow: Supplerende arkitektydelse for (1) energiparametre og (2) vægorientering. ....	17
Figur 2.1.3.2. Workflow: Håndtering og anvendelse af modtagne arkitektmodeller.....	19
Figur 2.1.3.3. Konceptuelt procesdiagram: Principielt designproces med fokus på redundans i den analyserede kontekst.....	20
Figur 2.1.3.4. Workflow: Indeklima-simulering, ventilationsprojektering, design- og projekteringsværdier .....	22
Figur 2.1.3.5. Workflow: Excel/Be10-link, performancedata og manuelle input i BIM.....	23
Figur 2.2.2.1. Isometriske skærbilleder af virtuel bygningsmodel.....	29
Figur 2.2.2.2. Planvisninger af testmodel. Den blå pil illustrerer modelleringsretning .....	30
Figur 2.2.2.3. Kvadranter og omregning af polære og retvinklede koordinater (MathIsFun.com, 2013) .....	31
Figur 2.2.3.1. Revit: Preview og indstillinger for gbXML eksport.....	32
Figur 2.2.3.2. Notepad++: gbXML-model som tekst. Eksempel for vindue (udsnit).....	33
Figur 2.2.3.3. Revit: Preview og indstillinger for gbXML eksport (Analytical Surfaces) .....	34
Figur 2.2.3.4. Revit-navngivning af BIM-objekter i testmodel eksporteret til gbXML .....	35
Figur 2.2.3.5. Opdeling af væg og vindue i eksporteret gbXML-model .....	36
Figur 2.2.4.1. Brugertilpasning af IFC-eksport i Revit 2015.....	38
Figur 2.2.4.2. Revit parameter med IfcGUID.....	39
Figur 2.2.4.3. Notepad++: IfcWindow repræsenteret som tekst (oprindeligt én linje) .....	39
Figur 2.2.4.4. DDS-CAD: Isometrisk visning af testmodel i IFC-format .....	39
Figur 2.2.4.5. IFC File Analyzer: Indstillinger anvendt i forbindelse med analysen .....	40
Figur 2.2.4.6. IFC File Analyzer/Excel: Antal af IfcWindow under summary .....	41
Figur 2.2.4.7. IfcProduct og relativ placering, princip efter (buildingSMART, 2013c) .....	41
Figur 2.2.4.8. Revit: Objektgeometri og skitserede IFC-værdier, lige væg .....	43
Figur 2.2.4.9. Revit: Objektgeometri og skitserede IFC-værdier, rund væg .....	45
Figur 2.2.4.10. Modelleringsretning: Wall01 og Wall02 .....	46
Figur 2.2.4.11. Wall 01: Retning og koordinatsystem for vinduesåbning.....	47
Figur 2.2.4.12. Wall 02: Retninger og koordinatsystemer for vindue og åbning.....	48
Figur 2.2.4.10. IFC File Analyzer/Excel: ObjectPlacement for vinduer i Wall 01 (øverst) og Wall 02 (nederst).....	49
Figur 2.2.4.13. IFC filen åbnet i ArchiCAD18 .....	49
Figur 2.2.5.1. Revit Lookup Add-In i User Interface .....	52
Figur 2.2.5.2. Revit Lookup: Snoop DB. Vinduer/døre under <i>FamilyInstance</i> .....	53
Figur 2.2.5.3. Snoop DB: 'Mark'-tag og relateret vindue.....	53
Figur 2.2.5.4. Snoop DB: Informationer for vindue (FamilyInstance).....	54
Figur 2.2.5.5. Snoop Current: Vindue (Element Class).....	55
Figur 2.2.5.6. Snoop Current: Vindue i ret facade (FamilyInstance / LocationPoint) .....	56
Figur 2.2.5.7. Revit: Objektgeometri og skitserede værdier (Location, Rotation, Facing orientation) .....	57
Figur 2.2.5.8. Snoop Current: Vindue i rund facade (FamilyInstance / LocationPoint) .....	58

Figur 2.2.5.9. Objektgeometri og skitserede værdier ved rund facade (Location, Rotation, Facing orientation) .....	59
Figur 2.2.5.10. Wall 01: Modelleringsretning og skitserede værdier (Location, Rotation, Facing orientation) .....	60
Figur 2.2.5.11. Wall 02: Modelleringsretning og skitserede værdier (Location, Rotation, Facing orientation) .....	61
Figur API13. Overfladeanalyse med <i>RaytraceBounce</i> (Conover, 2010, p. 18) .....	62
Figur API14. Skyggeanalyse af geometri (Conover, 2011, p. 18) .....	62
Figur API15. Revit Lookup: Transform information fra linket IFC-model læst af Revit API. De pågældende værdier tilhører vinduet i den runde facade. ....	63
Figur 2.2.6.1. BIM-link: Families med fejlmarkeringer .....	65
Figur 2.2.6.2. BIM-link: Kuldebroer og Arealer .....	66
Figur 2.2.6.3. Klimaskærm: Opvarmet etageareal som manuelt input .....	66
Figur 2.2.6.4. Gulvskema i Revit.....	67
Figur 2.2.6.5. Dæk i Rockwool Energy.....	67
Figur 2.2.6.6. Revit-planer vs. vinduesliste i Rockwool Energy .....	68
Figur oop1. Class/Object princip i C# (Greene & Stellman, 2013) .....	73
Figur 3.1.5.1. Visual C# template for Revit 2015 Add-In.....	76
Figur 3.1.5.2. Visual C#: Importerede Namespaces i template-genereret kode ( <i>Command Class</i> ) .....	76
Figur 3.1.5.3. Visual C#: Projekt-Namespace og basiskode Add-In typen <i>External Tool</i> .....	77
Figur 3.1.5.4. IntelliSense menu med forslag til <i>IExternalCommand</i> (Autodesk Inc., 2014j) .....	77
Figur 3.1.5.5. Visual C#: Template-genereret kode skaber adgang til API-data opbevaret i variablerne <i>app</i> og <i>doc</i> . ....	78
Figur 3.1.5.6. Visual C#: <i>FilteredElementCollector</i> med specifik kategori (vinduer) og type ( <i>FamilyInstance</i> ).....	79
Figur 3.1.5.7. Visual C#: Iteration med <i>foreach loop</i> .....	79
Figur 3.1.5.8. Visual Studio: Start Revit med Debug .....	80
Figur 3.1.5.9. Revit Lookup: <i>StorageType</i> for <i>Mark</i> -parameter.....	80
Figur 3.1.5.10. Visual C#: Kald til properties og parameter i <i>FamilyInstance</i> (vindue).....	81
Figur 3.1.5.11. Visual C#: Udtræk af areal-parameter .....	82
Figur 3.1.5.12. Revit Lookup: Areal parameter og værdi for vindue .....	82
Figur 3.1.5.13. Midlertidigt alternativ til API-opgivet vinduesareal.....	82
Figur 3.1.5.14. Visual C#: Beregning af orientering med <i>FacingOrientation</i> .....	83
Figur 3.1.5.15a. Basiskoordinater for vægbaseret vindue.    Figur 3.1.5.15b. Basiskoordinater for tagvindue. Pile angiver retningsvektor defineret af XYZ i <i>FacingOrientation</i> .....	84
Figur 3.1.5.16. Reference planer for top og bund af et tagvindue. ....	85
Figur 3.1.5.17. Tagvindue i modelkontekst og retningsvektor defineret af <i>FacingDirection</i> . ....	85
Figur 3.1.5.18. Visual C#: Beregning af hældning ud fra <i>FacingOrientation.Z</i> .....	86
Figur 3.1.5.19. Visual C#: <i>Worksheet</i> med modelnavn og start af stopur. ....	87
Figur 3.1.5.20. Visual C#: Navn, Id og Mark, skrives til Excel-kolonne 1, 4 og 5 .....	87
Figur 3.1.5.21. Visual C#: Afrundede loops, opsummeret output og afsluttende bekræftelse i Revit UI. ....	88
Figur 3.2.1. Aktivering af Add-In under External Tools, output til Excel og afsluttende dialogboks i Revit UI.....	89
Figur 3.2.2. Isometrisk billede af tilpasset testmodel Case2 .....	90
Figur 3.2.3.1. Vinduer, ' <i>Mark</i> '-tags og opmålte orienteringer i stueplan.....	92
Figur 3.2.3.2. Tagvinduer, ' <i>Mark</i> '-tags og opmålte hældninger i snit .....	93
Figur 3.2.4.1. Case 3: Isometrisk billede testet arkitektmodel (CUBO Arkitekter, 2013). ....	94

Figur 3.2.4.2. Case 3: Output med 107 vinduer til Excel (Ovalcirkel markerer selvstændigt ark for testmodellen).....	94
--	----

## Tabeller

Tabel 2.2.1. Beskrivelsesprincip for prioriterede Be10-parametre. Type repræsenterer parameterens primære association til BIM: Geometri (G) eller Egenskab (E) .....	27
Tabel 2.2.2. Relevante Be10-parametre for BIM-dataudtræk (repræsentativt udvalg) og markerede fokusparametre. ....	28
Tabel 2.2.3. Bygningsfakta om testmodel/sekundær case .....	30
Tabel 2.2.4.1. IFC File Analyzer/Excel: <i>ObjectPlacement</i> for væg.....	42
Tabel 2.2.4.2. IFC File Analyzer/Excel: <i>ObjectPlacement</i> for vinduesåbning .....	42
Tabel 2.2.4.3. IFC File Analyzer/Excel: <i>ObjectPlacement</i> for vindue.....	43
Tabel 2.2.4.4. IFA/Excel: <i>ObjectPlacement</i> for rund væg .....	44
Tabel 2.2.4.5. IFA/Excel: <i>ObjectPlacement</i> for åbning i rund væg .....	44
Tabel 2.2.4.6. IFC File Analyzer/Excel: <i>ObjectPlacement</i> for vindue i rund væg .....	44
Tabel 2.2.4.7. IFC File Analyzer/Excel: Wall 01 (øverst) og Wall 02 (nederst) .....	47
Tabel 2.2.4.8. IFC File Analyzer/Excel: <i>ObjectPlacement</i> for åbning i Wall 01 .....	47
Tabel 2.2.4.9. IFC File Analyzer/Excel: <i>ObjectPlacement</i> for åbning i Wall 02 .....	48
Tabel 3.1. Udviklingstrin i programmeringen .....	74
Tabel 3.2.1. Excel: Vinduesinformation oplistet i ark opkaldt efter model ( <i>her tm_ARK-S_02.rvt</i> ) .....	91
Tabel 3.2.3.1. Vinduesdata i Excel-ark, stueplan .....	92
Tabel 3.2.3.2. Data i Excel-ark, tagvinduer .....	93

## REFERENCER

Aalborg Universitet, 2011. *Aalborg-modellen for problembaseret læring*. [Online]  
Available at: <http://www.aau.dk/om-aau/aalborg-modellen-problembaseret-laering>  
[Senest hentet eller vist den 25 september 2012].

Aggerholm, S. & Grau, K., 2011. *Bygningers Energibehov*, Hørsholm: Statens Byggeforskningsinstitut, Aalborg Universitet.

Autodesk Inc., 2014a. *Help: Revit Fundamentals*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-71F2C8EE-2A90-4076-A6C7-702082566DDF>  
[Senest hentet eller vist den 27 september 2014].

Autodesk Inc., 2014b. *Help: Essential Skills*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-7F8CFFA4-22CB-43CA-84EA-332A27A0A0F0>  
[Senest hentet eller vist den 27 september 2014].

Autodesk Inc., 2014c. *Help: Schedules*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-F50D6FF4-859E-43A2-A2F6-81C84A1BA0EB>  
[Senest hentet eller vist den 7 oktober 2014].

Autodesk Inc., 2014d. *Help: Export*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-A5E61A08-9635-44D0-93CA-75C15282119B>  
[Senest hentet eller vist den 7 oktober 2014].

Autodesk Inc., 2014e. *Help: Energy Analysis using Building Elements - Model Validation*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-9DB01490-A0A1-4DA2-A9A9-4F766958017A>  
[Senest hentet eller vist den 8 oktober 2014].

Autodesk Inc., 2014f. *Help: Linked Models*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-0FBC74D9-C739-4ED3-962E-20DC4526A678>  
[Senest hentet eller vist den 15 oktober 2014].

Autodesk Inc., 2014g. *Help: Tools and Techniques*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-60C4C656-52F9-4A95-8A5C-028011CF0397>  
[Senest hentet eller vist den 27 oktober 2014].

Autodesk Inc., 2014h. *Revit 2015 API [RevitAPI.chm in Revit Software Development Kit]*, s.l.: Autodesk Inc..

Autodesk Inc., 2014i. *Autodesk Developer Network - Autodesk Revit*. [Online]  
Available at: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975>  
[Senest hentet eller vist den 10 november 2014].

Autodesk Inc., 2014j. *Help: Revit API Developers Guide*. [Online]  
Available at: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-F0A122E0-E556-4D0D-9D0F-7E72A9315A42>

[Senest hentet eller vist den 29 november 2014].

Autodesk Inc., 2014k. *My First Revit Plug-in Training*. [Online]

Available at: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=16777469>

[Senest hentet eller vist den 2 december 2014].

Autodesk Inc., 2014m. *Revit 2015 API [RevitAPI.chm in Revit Software Development Kit]*, s.l.: Autodesk Inc..

Bejder, E. & Olsen, W., 2007. *Anlægsteknik 2 - Styling af byggeprocessen*. 2. red. Kgs. Lyngby: Polyteknisk Forlag.

Beyer, H. & Holtzblatt, K., 1998. *Contextual Design : Defining Customer-Centered Systems*. San Francisco, CA: Morgan Kaufmann Publishers.

BIM AARHUS, 2012. *Information | BIM AARHUS*. [Online]

Available at: <http://bimaarhus.dk/information/>

[Senest hentet eller vist den 28 september 2014].

BLIS, 2014. *IFC Solutions Factory*. [Online]

Available at: <http://www.blis-project.org/IAI-MVD/>

[Senest hentet eller vist den 5 november 2014].

buildingSMART, 2013a. *Home - Welcome to buildingSMART-Tech.org*. [Online]

Available at: <http://www.buildingsmart-tech.org/>

[Senest hentet eller vist den 15 oktober 2014].

buildingSMART, 2013b. *IFC Roadmap Summary*. [Online]

Available at: <http://www.buildingsmart-tech.org/future-extensions/ifc-roadmap>

[Senest hentet eller vist den 15 oktober 2014].

buildingSMART, 2013c. *Product Placement*. [Online]

Available at: <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/templates/product-placement.htm>

[Senest hentet eller vist den 5 november 2014].

buildingSMART, 2013d. *IfcLocalPlacement*. [Online]

Available at: <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcgeometricconstraintresource/lexical/ifclocalplacement.htm>

[Senest hentet eller vist den 5 november 2014].

buildingSMART, 2014. *Open Standards 101 | buildingSMART*. [Online]

Available at: <http://www.buildingsmart.org/standards/technical-vision/open-standards-101/>

[Senest hentet eller vist den 4 december 2014].

CASE Design, 2014. *CASE APPS*. New York, NY: CASE Design, Inc..

Conover, S., 2010. *Analyze Geometry of Buildings Using the Autodesk® Revit® API*. s.l.:Autodesk University.

Conover, S., 2011. *Geometric Progression: Further Analysis of Geometry using the Autodesk Revit 2012 API*. s.l.:Autodesk University.

CUBO Arkitekter, 2013. *Case 3: Revit-arkitektmodel*. Aarhus: CUBO ARKITEKTER A/S.

Dalux, 2012. *Rockwool Energy Design - Revit integration*. [Online]  
Available at: <https://www.youtube.com/watch?v=3PRMP0y95KU>  
[Senest hentet eller vist den 6 november 2014].

Dansk Standard, 2011. *DS418:2011 Beregning af bygningers varmetab*. 7. red. Charlottenlund: Dansk Standard.

Data Design System, 2014. *DDS-CAD Viewer*. [Online]  
Available at: <http://www.dds-cad.net/downloads/dds-cad-viewer/>  
[Senest hentet eller vist den 10 oktober 2014].

Eastman, C., Teicholz, P., Sacks, R. & Liston, K., 2011. *BIM Handbook - A guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*. 2. red. New Jersey: John Wiley & Sons Inc..

Energistyrelsen, 2014. *Bygningsreglementet 01.01.2014 - BR10*. [Online]  
Available at: <http://bygningsreglementet.dk/br10/0/42>  
[Senest hentet eller vist den 2 oktober 2014].

EQUA, 2014. *IDA ICE - EQUA - Building and Tunnel simulations*. [Online]  
Available at: <http://www.equa.se/index.php/en/ida-ice>  
[Senest hentet eller vist den 20 oktober 2014].

Gonçalves, A., 2014. *DevTV: Introduction to Revit Programming*, s.l.: Autodesk Inc..

Greene, J. & Stellman, A., 2013. *Head First C#*. 3. red. Sebastopol, CA: O'Reilly Media.

Hildebrandt, S., 2011. *Steen Hildebrandt - En organisationsmodel*. [Online]  
Available at: <http://www.steenhildebrandt.dk/2011/09/en-organisationsmodel/>  
[Senest hentet eller vist den 2 oktober 2014].

Ho, D., 2014. *Notepad++ Home*. [Online]  
Available at: <http://notepad-plus-plus.org/>  
[Senest hentet eller vist den 26 september 2014].

IES, 2014. *Integrated Environmental Solutions*. [Online]  
Available at: <http://www.iesve.com/>  
[Senest hentet eller vist den 20 oktober 2014].

Kim, J. B. et al., 2014. Developing a physical BIMlibrary for building thermal energy simulation. *Automation in Construction* 50 (2015), 21 oktober, pp. 16-28.

Klima-, Energi- og Bygningsministeriet, 2013. *Bekendtgørelse om anvendelse af informations- og kommunikationsteknologi (IKT) i offentligt byggeri - retsinformation.dk*. [Online]  
Available at: <https://www.retsinformation.dk/Forms/R0710.aspx?id=145421&exp=1>  
[Senest hentet eller vist den 20 december 2014].

Leavitt, H. J. & Bahrami, H., 1988. *Managerial Psychology: Managing Behavior in Organizations*. 5. red. Chicago, IL: The University of Chicago Press.



- Lipman, R., 2014. *NIST - IFC File Analyzer*. [Online]  
Available at: <http://www.nist.gov/el/msid/infotest/ifc-file-analyzer.cfm>  
[Senest hentet eller vist den 10 oktober 2014].
- MathIsFun.com, 2013. *Polar and Cartesian Coordinates*. [Online]  
Available at: <http://www.mathsisfun.com/polar-cartesian-coordinates.html>  
[Senest hentet eller vist den 10 oktober 2014].
- Microsoft, 2012a. *Visual C# IntelliSense*. [Online]  
Available at: [http://msdn.microsoft.com/en-us/library/43f44291\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/43f44291(v=vs.110).aspx)  
[Senest hentet eller vist den 2 december 2014].
- Microsoft, 2012b. *Math.Atan2 Method (System)*. [Online]  
Available at: [http://msdn.microsoft.com/en-us/library/vstudio/system.math.atan2\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/system.math.atan2(v=vs.100).aspx)  
[Senest hentet eller vist den 3 december 2014].
- MOE, 2014a. *Firmaprofil 2014*, København: MOE A/S Rådgivende Ingeniører.
- MOE, 2014b. *MOE - Byggeri | Energi | Infrastruktur*. [Online]  
Available at: <http://moe.dk/>  
[Senest hentet eller vist den 24 september 2014].
- MOE, 2014c. *BIM+Be10*. Aarhus/online: BIM AARHUS.
- Mondrup, T. F., Karlshøj, J. & Vestergaard, F., 2012. *INFORMATION DELIVERY MANUALS TO FACILITATE IT SUPPORTED ENERGY ANALYSIS*, Kgs. Lyngby: Technical University of Denmark.
- NIST, 2014. *National Institute of Standards and Technology - IFC File Analyzer*. [Online]  
Available at: <http://www.nist.gov/el/msid/infotest/ifc-file-analyzer.cfm>  
[Senest hentet eller vist den 10 oktober 2014].
- NTI, 2014. *NTI CADcenter*. [Online]  
Available at: <http://www.nti.dk/>  
[Senest hentet eller vist den 2 oktober 2014].
- ROCKWOOL, 2013. *Rockwool Energy Design 4.0*. [Online]  
Available at: <http://www.rockwool.dk/beregninger/energiberegning>  
[Senest hentet eller vist den 22 september 2014].
- SBi, 2013. *Simulering med BSim*. [Online]  
Available at: <http://www.sbi.dk/indeklima/simulering>  
[Senest hentet eller vist den 7 oktober 2014].
- Stumpf, A. L., Kim, H. & Jenicek, E. M., 2011. *Early Design Energy Analysis Using Building Information Modeling Technology*, Champaign, IL: US Army Engineer Research and Development Center.
- Tammik, J., 2012d. *The Building Coder: Exporting Parameter Data to Excel, and Re-importing*. [Online]  
Available at: <http://thebuildingcoder.typepad.com/blog/2012/09/exporting-parameter-data-to-excel.html>  
[Senest hentet eller vist den 15 december 2014].
- Tammik, J., 2013e. *The Building Coder*. [Online]  
Available at: <http://thebuildingcoder.typepad.com/blog/2013/09/boolean-operations-for-2d->

[polygons.html](#)

[Senest hentet eller vist den 29 december 2014].

Tammik, J., 2014a. *The Building Coder*. [Online]

Available at: <http://thebuildingcoder.typepad.com/blog/2014/05/new-revit-2015-sdk-samples.html>

[Senest hentet eller vist den 10 november 2014].

Tammik, J., 2014a. *The Building Coder*. [Online]

Available at: <http://thebuildingcoder.typepad.com/blog/>

[Senest hentet eller vist den 1 november 2014].

Tammik, J., 2014c. *The Building Coder: Add-In Wizards for Revit 2015*. [Online]

Available at: <http://thebuildingcoder.typepad.com/blog/2014/05/add-in-wizards-for-revit-2015.html>

[Senest hentet eller vist den 1 december 2014].

Teknologisk Institut, 2014. *Bæredygtigt Byggeri - Ydelser - Teknologisk Institut*. [Online]

Available at: <http://www.teknologisk.dk/ydelser/baeredygtigt-byggeri/30894>

[Senest hentet eller vist den 23 september 2014].

van Kesteren, A., 2014. *Encoding Standard*. [Online]

Available at: <https://encoding.spec.whatwg.org/>

[Senest hentet eller vist den 22 december 2014].

Velez, A., 2014. *IFC Exporter for Revit*. [Online]

Available at: <http://sourceforge.net/projects/ifcexporter/>

[Senest hentet eller vist den 20 oktober 2014].

VELUX, 2014. *FAQ | VELUX*. [Online]

Available at: <http://www.velux.dk/private/produkter/faq>

[Senest hentet eller vist den 20 december 2014].

# APPENDIKS A

---

Emne: Opstartsmøde

Dato: 2014.09.22

Deltagere:

SEM  
TOD  
RLJ  
KS  
PLJ

---

Referat:

*Noter angives med kantet parentes: [Note]*

- Dialog om problemstilling, muligheder og interesser.
- Virksomhedens nuværende optimeringsinitiativ.
  - Be10-relaterede skemaer oprettet i Energi-template for Revit-projekt. Skemaer eksporteres til Excel.
  - Egen API-løsning linker Excel og Be10
  - Vurdering af vinduesplacering/lokalisering er besværlig og er løst med en *Work Around*, hvor et midlertidigt rum modelleres uden for bygningens facader, så 'facing x' kan læses som relation i Revit-skema.
- I de fleste tilfælde bestemmes nogle værdier på baggrund af en menneskelig vurdering. Det er umuligt at gøre alting målbart, men målbare data er man interesseret i, at overføre så effektivt, som muligt. Det er dog stadigvæk relevant at se på muligheder for automatisering af eksempelvis beregninger af skygger og dagslysfaktorer.
- Andre mulige måder at optimere med andre metoder/værktøjer.
  - Ikke sikkert nuværende indgangsvinkel er den optimale, men har resulteret i et tidsforbrug, som er mere end halveret i forhold til en traditionel opmålingsproces. Muligheder og potentiale skal afdækkes, så der er beslutningsgrundlag for videre udvikling af løsning.
  - En grundig analyse af metoder er værdifuld og en konkluderet kvalificeret løsning kan evt. være projektets produkt ift. den givne tidsramme.
  - Data skal ikke nødvendigvis overføres direkte til Be10. Det kan lige så vel være til Excel. **Det vigtigste er at man udtrækker data på en struktureret måde, som giver et ensartet output i hvert projekt.** Så kan man f.eks. bruge Excel templates til sortere data efterfølgende.
  - Som regel vil ingeniøren måske endda foretrække dataudtræk til regneark, hvor det er muligt at udføre mellemregninger, som er forbundet med de menneskelige vurderingstilfælde.
- Perspektiv
  - Interoperabilitet generelt. Føre beregnede værdier tilbage i BIM og anvende data i andre analyse-/simuleringsværktøjer.
- Næste skridt
  - Undersøge kontekst og eksisterende løsninger [Energianalyse med data fra BIM]

# APPENDIKS B

---

Emne: Projekt møde

Dato: 2014.10.03 + 11.04

Deltagere:

SEM  
PLJ

---

Referat:

*Noter angives med kantet parentes: [Note]*

## INFORMATIONSHÅNDTERING

- Gennemgang af 1. udkast til flowmodel for kontekst [iht. Slides fra BIM Aarhus, feb. 2014.]
- **Q:** *Hvordan håndterer I aktiviteten med at tilføje informationer til vinduer?* [Slide 11]  
[Ikke teknisk muligt at skrive til linkede modeller, så information må tilføjes i lokal kopi af arkitektmodel. Tilføjet information eksisterer kun lokalt, ikke i overordnet projektmateriale. Aktiviteten må således gentages ved hver udveksling af fagmodeller og gør processen redundant.]  
  
**A:** Det er arkitekten der tilføjer informationerne. Ingeniørerne sørger for at de supplerende ydelser skrives ind i den projektspecifikke IKT-aftale. Arkitektmodel linkes ind i energitemplate, hvor spaces og zones modelleres. I den forbindelse udleveres et kompendie, hvor proceduren er beskrevet. [Kompendier udleveret - beskriver at arkitekt tilføjer og ingeniører kopierer yderligere]
- De erfarne ingeniører kan have et begrænset omfang af BIM-relaterede kompetencer. Derfor har nogle medarbejdere en funktion som CAD-operatører. De sørger for den nødvendige opsætning i den nuværende procedure.

## WORKFLOW: ROLLER OG ANSVAR

- **Q:** *Hvordan er de ingeniørfaglige opgaver fordelt ift. indeklima og dagslys. Og hvilke parametre kan de have tilfælles?* [I kontekst af energiramme]

**A:** Energiberegning, indeklima og dagslys af den samme [Energi- og indeklimaingeniøren]. Og ellers konstruktioner og ventilation som vist i modellen. Ventilationsingeniøren bruger bl.a. designværdier for luftmængder, som udgangspunkt i ventilationsprojekteringen.

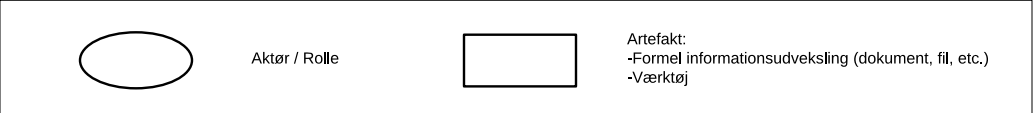
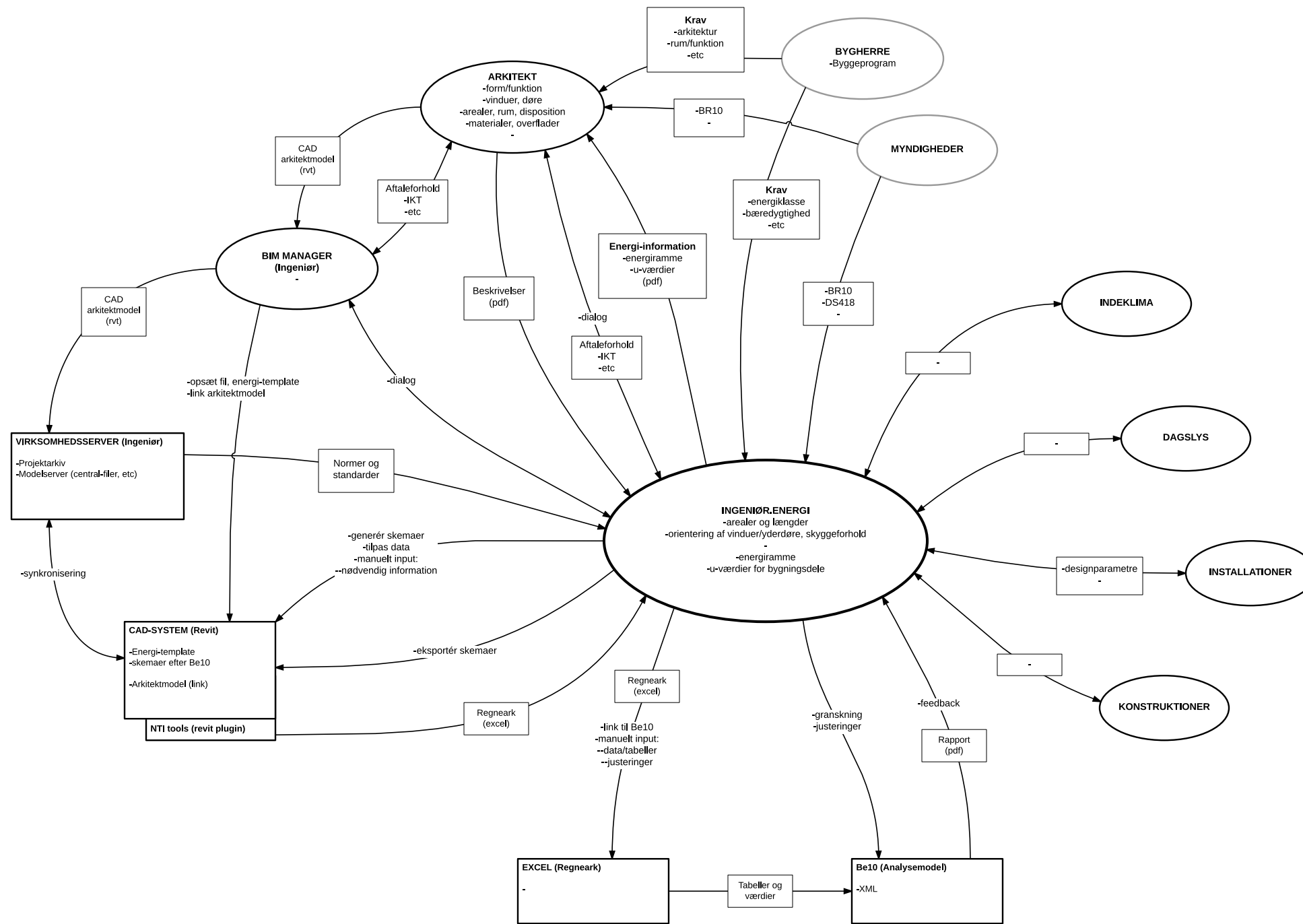
- **Q:** *Hvilke analyseværktøjer er forbundet parametrene [fx luftmængder] og i hvilket omfang kan de indgå?* [Iterativt i forhold til designproces, f.eks. flere faser]

**A:** Energi- og indeklimaingeniør udfører en **indeklimasimulering** med **BSim**, på baggrund af informationer om bl.a. person- og inventarbelastning, rumgeometri og type. [Input fra byggeprogram, arkitektprojekt, etc.]. Det resulterer i nogle designværdier for bl.a. luftmængder, som føres videre til ventilationsingeniøren. [Evt. i egenskabsdata for spaces/zoner i Revit] Ventilationsprojekteringen giver nogle projekteringsværdier, som kan være forskellige fra designværdierne. Dem bruger indeklima- og energiingeniøren i en ny simulering [BSim] til at kontrollere om værdierne er tilstrækkelige eller der må foretages justeringer. Processen kan principielt gå om, hver gang der er lavet ændringer i arkitektprojektet.

- **Dagslysfaktor** bestemmes ved **beregning**.
- Termisk indeklima, energi og dagslys er indbyrdes afhængige og bearbejdes derfor, så vidt muligt, samtidigt.
- Supplerende
  - o Arkitekterne kan bruge mange forskellige begreber i rumnavne. [Fx mødelokale/seminarrum eller wc/toilet, etc.] For indeklima/energi er det funktionen, der er vigtig. Ingeniøren vurderer rummene i forhold til en bestemt liste af rumtyper/spaces. Det kan være relevant at effektivisere den aktivitet, i forhold til at udpege (1) rum som space (2) spaces i en given zone – eksempelvis med en funktion svarende til formatpensel.

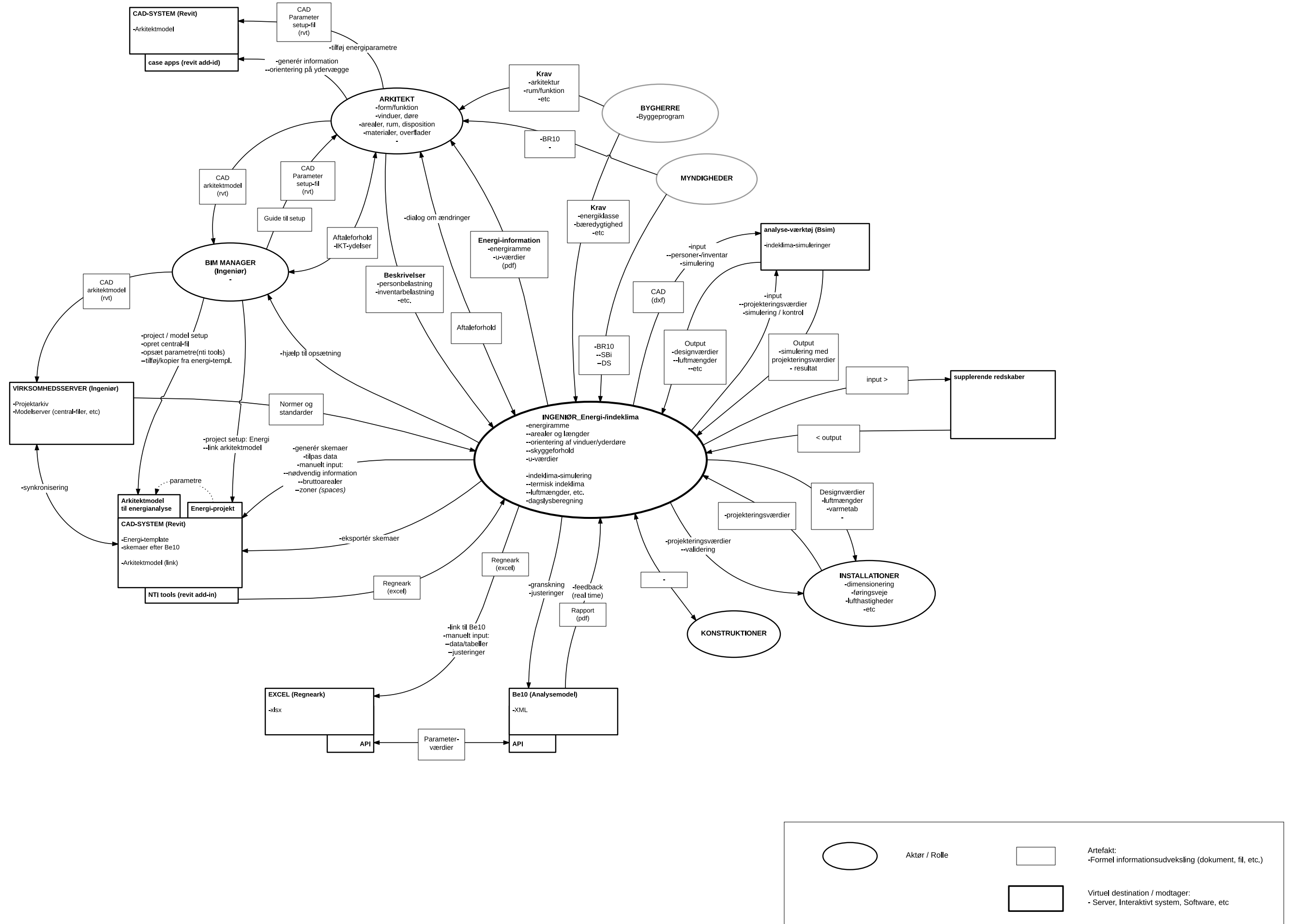
# APPENDIKS C

Flow model: Energianalyse  
2014.10.02



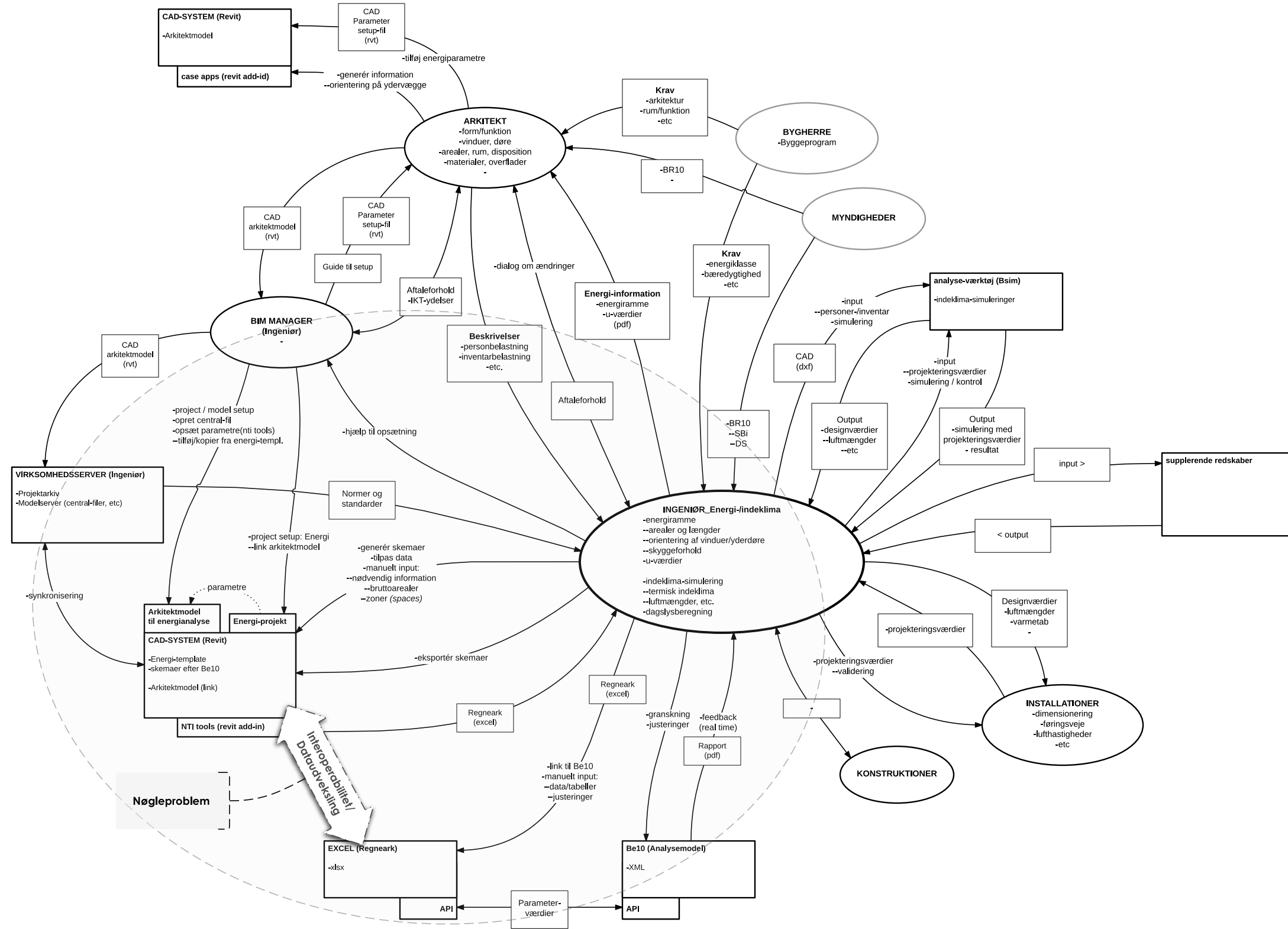
# APPENDIKS C

Flow model: Kontekst v.2  
2014.10.07



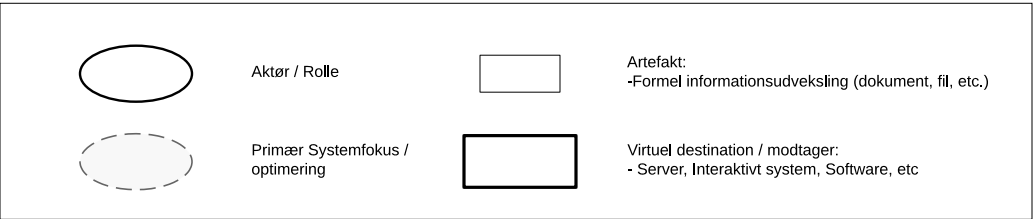
# APPENDIKS C

Flow model: Kontekst v.2  
2014.10.07



Nøgleproblem

Interoperabilitet/  
Dataudveksling





# APPENDIKS D

Emne: Prioriterede Be10 parametre.

Noter angives med kantet parentes: [Note]

## PRINCIP

[Overordnet Be10 kategori]				
Uddata	Inddata	Be10 parameter	Note	
	x	[Parameter]	[Type]	
x		[Parameter]	[Type]	

[Type = Parameterens primære association:

Geometri: G og/eller Egenskab: E ]

Uddata: Input i BIM - fra Be10

Inddata: Input i Be10 - fra BIM

Parameter og prioritet:

- Høj: 1
- Middelhøj: 2
- Ubestemt: Ingen markering
- *Relevant i fremadrettet perspektiv: Kursiv*

## PARAMETRE

Bygning				
U	I	Be10 parameter	Note	
		2 Opvarmet etageareal	G	
		2 Opvarmet kælder	G	
Ydervægge, tage og gulve				
U	I	Be10 parameter	Note	
		2 Transmissionsareal	G	
		<i>Transmissionskoefficient (U)</i>	E	
Fundamenter og samlinger ved vinduer				
U	I	Be10 parameter	Note	
		2 Kuldebrolængde	G	
		2 Linjetab	G	

Vinduer og yderdøre			
U	I	Be10 parameter	Note
		<b>1 Antal</b>	G <b>Topprioritet</b>
		<b>1 Orientering</b>	
		<b>1 Hældning</b>	
		<b>1 Transmissionsareal</b>	
		<i>Transmissionskoefficient (U)</i>	E
		Glasandel	G
		Rudens solvarmetransmittans	E
		<i>2 Skygger</i>	G [Separat skema ↓]
		Solafskærmning	G
Skygger			
		Be10 parameter	Note
		<i>Horisont</i>	G
		<i>Udhæng</i>	G
		<i>Skygge til højre</i>	G
		<i>Skygge til venstre</i>	G
		<i>Horisont</i>	G
Uopvarmede rum			
		Be10 parameter	Note
		Bruttoareal	G
		Ventilationstab	E
Uopvarmede rum - transmissionstab fra bygning			
		Be10 parameter	Note
		Areal	G
		Transmissionskoefficient (U)	E
Uopvarmede rum - transmissionstab fra omgivelserne			
		Be10 parameter	Note
		Areal	G [Evt. udtræk af areal mod jord]
		Transmissionskoefficient (U)	E

Ventilation: Excel → Revit og anvendelse i ventilationsprojektering			
U	I	Be10 parameter	Note
		Areal	G
		Driftstid	E
		Mekanisk ventilation om vinteren i brugstiden [l/s m <sup>2</sup> ]	E
		Temperaturvirkningsgrad	E
		Indblæsningstemperatur	E
		Elvarmeflader	E
		Naturlig ventilation om vinteren i brugstiden	E
		Specifikt elforbrug til lufttransport	E
		Mekanisk ventilation om sommeren i brugstiden	E
		Naturlig ventilation om sommeren i brugstiden	E
Internt varmetilskud			
U	I	Be10 parameter	Note
		2 Areal	G
		2 Personer [W/m <sup>2</sup> ]	E ARK/BYGH
		2 Apparat i brugstiden[W/m <sup>2</sup> ]	E
		2 Apparat udenfor brugstiden[W/m <sup>2</sup> ]	E
Belysning			
U	I	Be10 parameter	Note
		Areal	G
		Eleffekt til almenbelysning i brugstiden, minimum	E
		Installeret effekt til almenbelysning i brugstiden	E
		Belysningsniveau [lux]	E
		Dagslysfaktor [%]	GE
		Dagslysstyring	E
		Benyttelsesfaktor	E
		Eleffekt til arbejdslamper i brugstiden [W/m <sup>2</sup> ]	E
		Eleffekt til anden belysning i brugstiden	E
		Standby effekt til almenbelysning udenfor brugstiden	E
		Eleffekt til belysning udenfor brugstiden	E

# APPENDIKS E

## 1.1 KONFIGURATION AF REVIT SDK OG SAMPLES

### 01 Microsoft Visual Studio

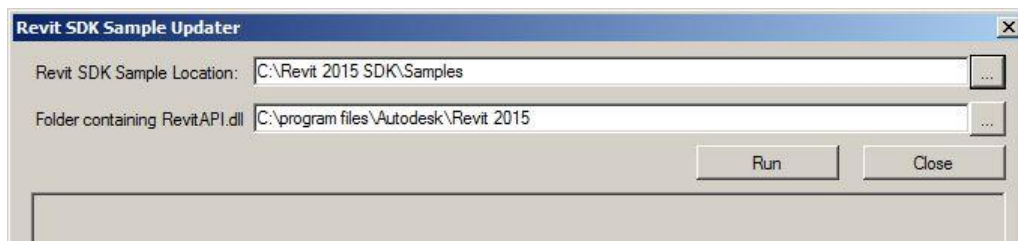
Anvendt i dette tilfælde

### 02 Installere SDK

(dl: url: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2484975>)

### 03 Opdatere assembly paths

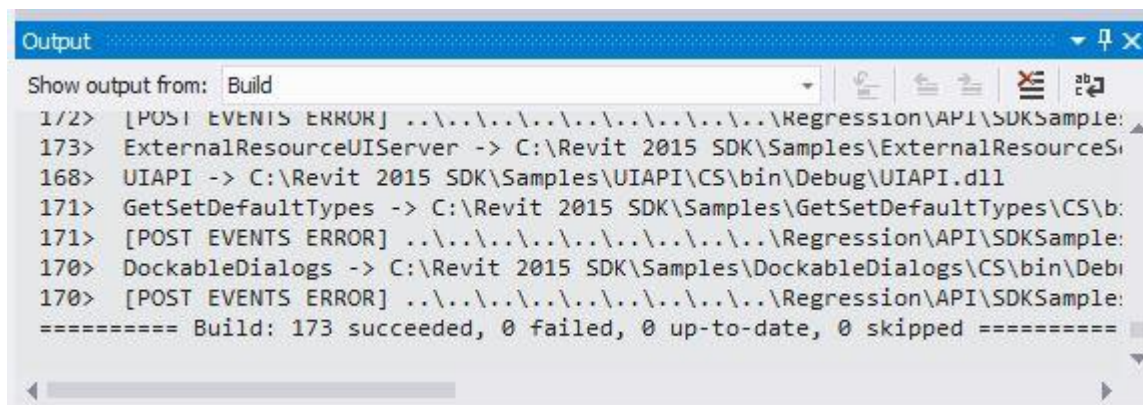
*RevitAPIDllsPathUpdater.exe*



### 04 Compile SDK

File > open > project/solution: ..\Revit 2015 SDK\Samples\SDKSamples2015.sln

Build > Build solution:



## 05 Opsæt RvtSamples

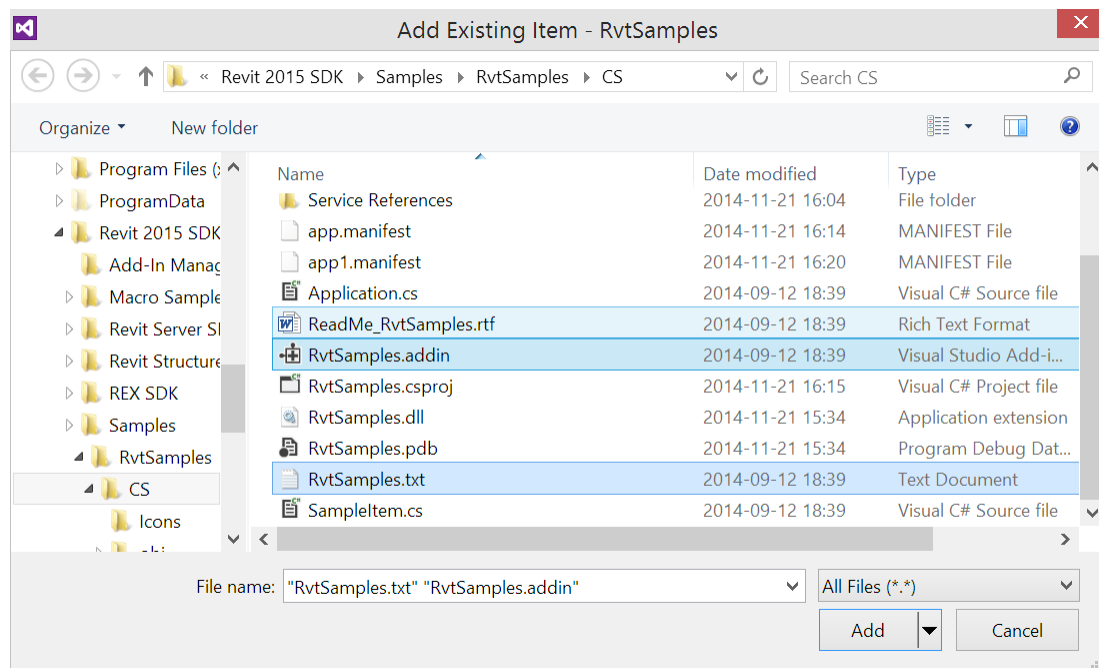
Manifest: <http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-4FFDB03E-6936-417C-9772-8FC258A261F7>

En add-in er først tilgængelig i Revit, når den er registreret. Det sker vha. en såkaldt manifest som fortæller Revit, hvilken DLL den skal læse og hvor. Den korrekte sti skal derfor skrives ind i manifesten.

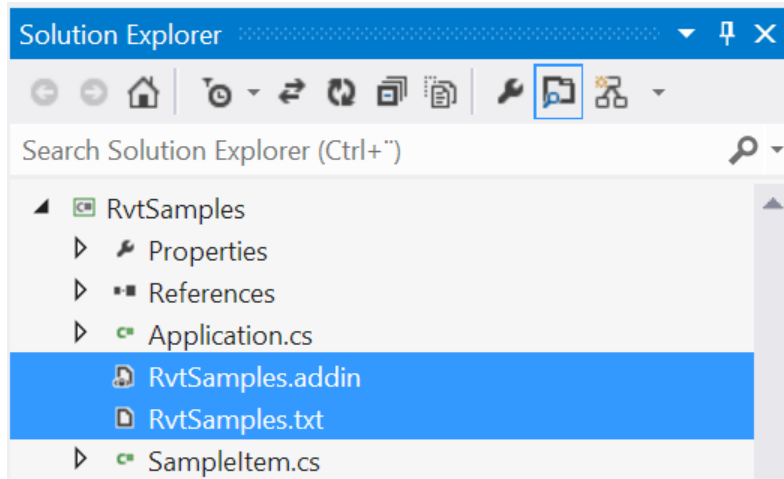
RvtSamples er en add-in, der fungerer som indholdsfortegnelse for de resterende samples. Den gør det muligt at teste de forskellige samples, uden at installere hvert enkelt. Det sker gennem en tekstfil der refererer til sample's manifest.

Hvis ikke SDK er installeret i default location (C:\Revit 2015 SDK) er det nødvendigt at tilføje den rigtige sti i tekstfilen RvtSamples.txt. Både manifest (RvtSamples.addin) og tekst-fil kan med fordel tilføjes i Visual Studio og rettes her fra.

Project > add existing item:



I denne undersøgelse er SDK installeret til *default* location, så tekstfilen rettes ikke. Det gør manifesten derimod.



Ændre manifest, RvtSamples.addin. Skal pege på kompileret .dll fil:

C:\Revit 2015 SDK\Samples\RvtSamples\CS\RvtSamples.dll

```
RvtSamples.addin  X
<?xml version="1.0" encoding="utf-8"?>
<RevitAddIns>
  <AddIn Type="Application">
    <Name>External Tool</Name>
    <Assembly>C:\Revit 2015 SDK\Samples\RvtSamples\CS\RvtSamples.dll</Assembly>
    <ClientId>42cb0a70-2ee7-4e64-a42d-87b9cdcc41c8</ClientId>
    <FullClassName>RvtSamples.Application</FullClassName>
    <VendorId>ADSK</VendorId>
    <VendorDescription>Autodesk, www.autodesk.com</VendorDescription>
  </AddIn>
</RevitAddIns>
```

## 06 Register add-in

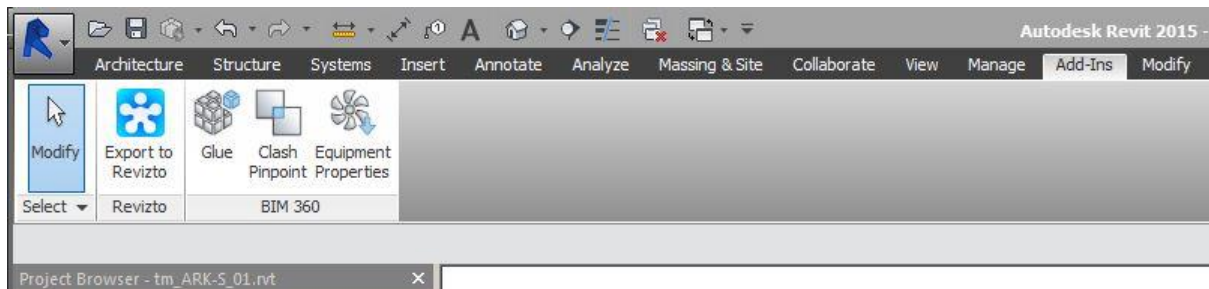
<http://help.autodesk.com/view/RVT/2015/ENU/?guid=GUID-4FFDB03E-6936-417C-9772-8FC258A261F7>

Gem manifest i/kopiér til:

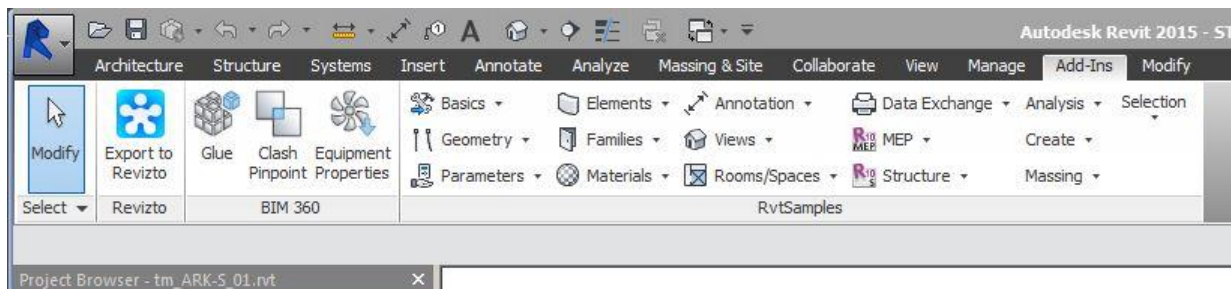
- *Ikke-brugerspecifik*: C:\ProgramData\Autodesk\Revit\Addins\2015\
- *Brugerspecifik*: C:\Users\\AppData\Roaming\Autodesk\Revit\Addins\2015\

Start Revit.

Før:



Efter:



Efterfølgende konfigureres Revit Lookup.

## 1.2 REVIT LOOKUP

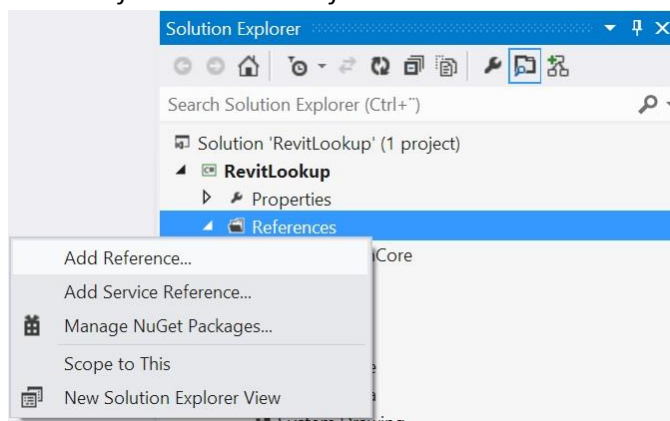
Kildekode på GitHub: <https://github.com/jeremytammik/RevitLookup>

### 1.2.1 VISUAL STUDIO:

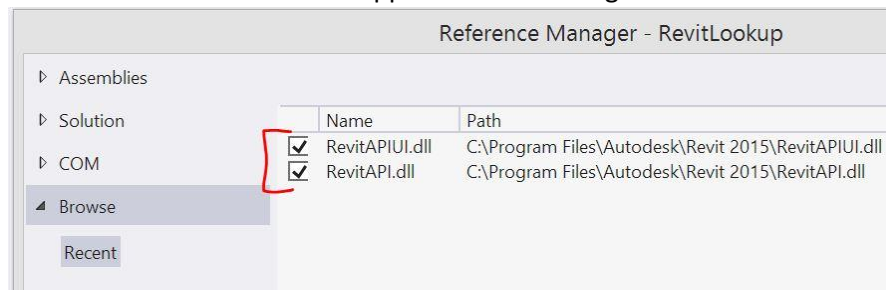
- Open > Project/Solution: /<sti>/RevitLookup.sln

#### Solution explorer:

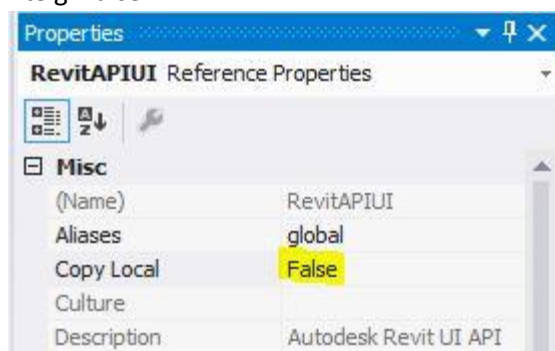
- Udfold *References*, find RevitAPI og RevitAPIUI, kontrollér *Properties/Path*
  - Skal svare pege på de respektive filer i installationsmappen for Revit
- Ingen/forkert path: H-klik > Remove begge.
  - H-klik: *References* > *Add Reference* i kontekst menu:



- Browse > Revit installationsmappe > RevitAPI.dll og RevitAPIUI.dll > OK



- Korrekt path: Se '*Copy Local*'
  - Vælg '*False*'





**Build > Rebuild All**

```

1>----- Rebuild All started: Project: RevitLookup, Configuration: Debug (For Release b
1> RevitLookup -> C:\DevLib\Res\Bc\RevitLookup-2015.0.0.2\CS\bin\Debug\RevitLookup.dll
1>         1 file(s) copied.
1>         1 file(s) copied.
===== Rebuild All: 1 succeeded, 0 failed, 0 skipped =====

```

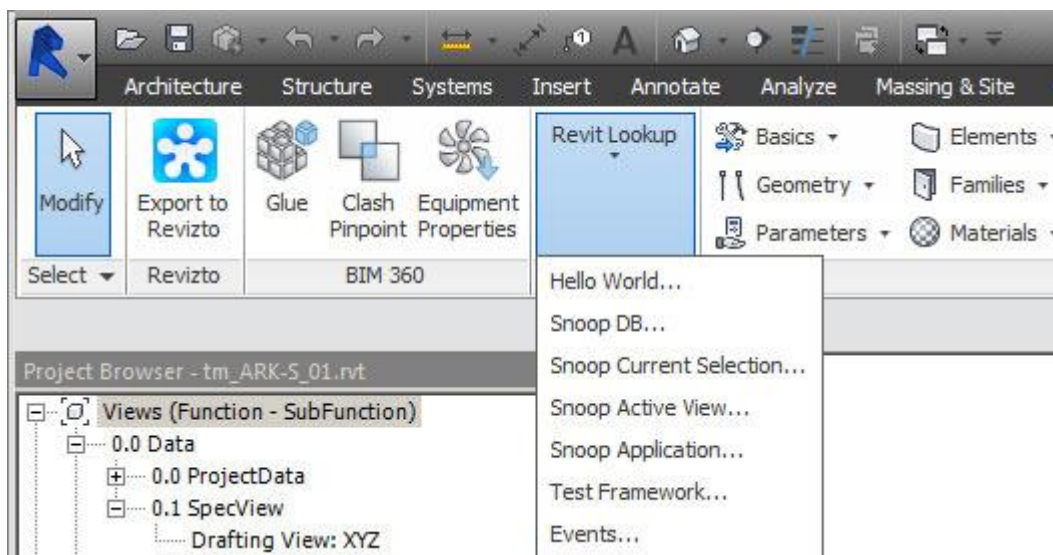
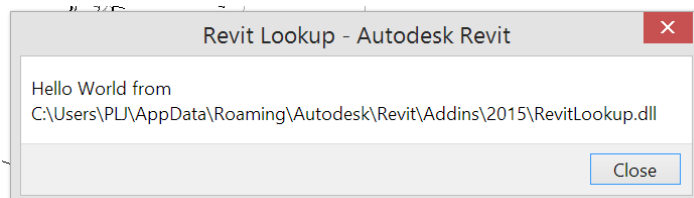
*Build Output* viser at den kompilerede DLL er placeret under \bin\Debug\. Dette skal skrives ind i manifest.

```
<Assembly>C:\DevLib\Res\Bc\RevitLookup-2015.0.0.2\CS\bin\Debug\RevitLookup.dll</Assembly>
```

Ligesom ved RvtSamples, gemmes manifesten i addins mappen (C:\ProgramData\Autodesk\Revit\Addins\2015\)

**Start Revit**

RevitLookup er registreret:

**Hello World...**

# APPENDIKS F

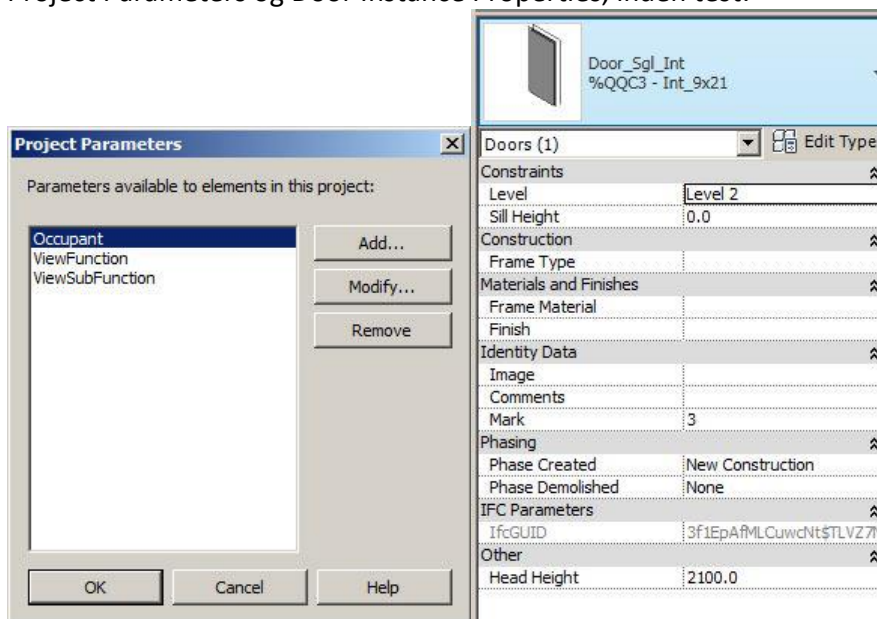
## 1.1 FORSØG MED 'FIRERATING'-SAMPLE

Sample består af 3 commands/brugeraktiverede handlinger

- Apply parameter
- Export to Excel
- Import from Excel

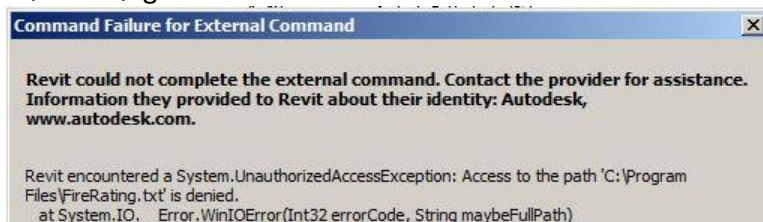
### Revit

Project Parameters og Door Instance Properties, inden test:



## 1.2 APPLY PARAMETER

Hvis man uden videre forsøger at starte i Revit (parameters > Apply Parameter) bliver man mødt af følgende besked:



Koden peger på 'C:\Program Files\FireRating.txt'. Windows kræver administratorrettigheder for at skrive i mappen og filen findes ikke, så det er ikke underligt, at fejlen meldes. Tid til at se på samplekoden.

### 1.2.1 SDK: \SAMPLES\FIRERATING\VB.NET\

Dette sample er skrevet i VB, modsat hovedparten af de øvrige samples, som overvejende er i C#.

### 1.2.2 VISUAL STUDIO:

File > Open > Project/Solution > FireRation.vbproj

MainModule.vb:

Starter med en halv kryptisk kommentar

```

Module MainModule
    ' adjust the path if you save this program in the different location.
    Public Const mPath As String = "..\..\.."
    Public Const mFilename As String = "FireRating.txt"
    Public Const mExcelFilename As String = "FireRating.xls"
    Public Const mGroupName As String = "Fire"
    Public Const mParameterName As String = "Fire Rating"
    Public Const mCategoryName As String = "Doors"
End Module

```

Jeg har i baghovedet at det er et sample, så forfatteren har ikke nødvendigvis haft intentioner om, at sikre OOTB funktionalitet. (F.eks. at det finder frem til den sharedparam fil, der reelt er loaded)

Jeg prøver at hjælpe lidt ved at oprette en mappe til formålet og justere 'path' i koden. Stien formodes at tage udgangspunkt i C:\Program Files\Autodesk\Revit 2015

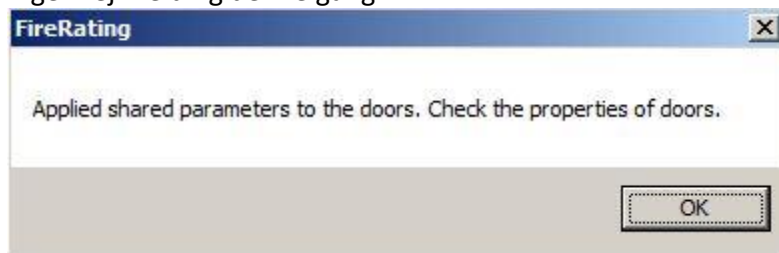
Den skal således gå 3 trin tilbage, til C:\  
Stien rettes til:

```
Public Const mPath As String = "..\..\..\DevLib\CSTRVT"
```

**Rebuild FireRating.vbproj**

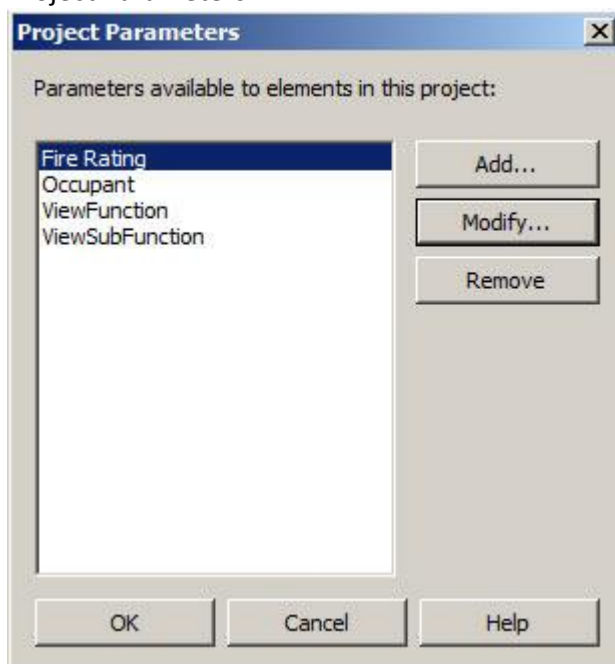
## 1.2.3 REVIT FORSØG 2

Ingen fejlmedling denne gang:

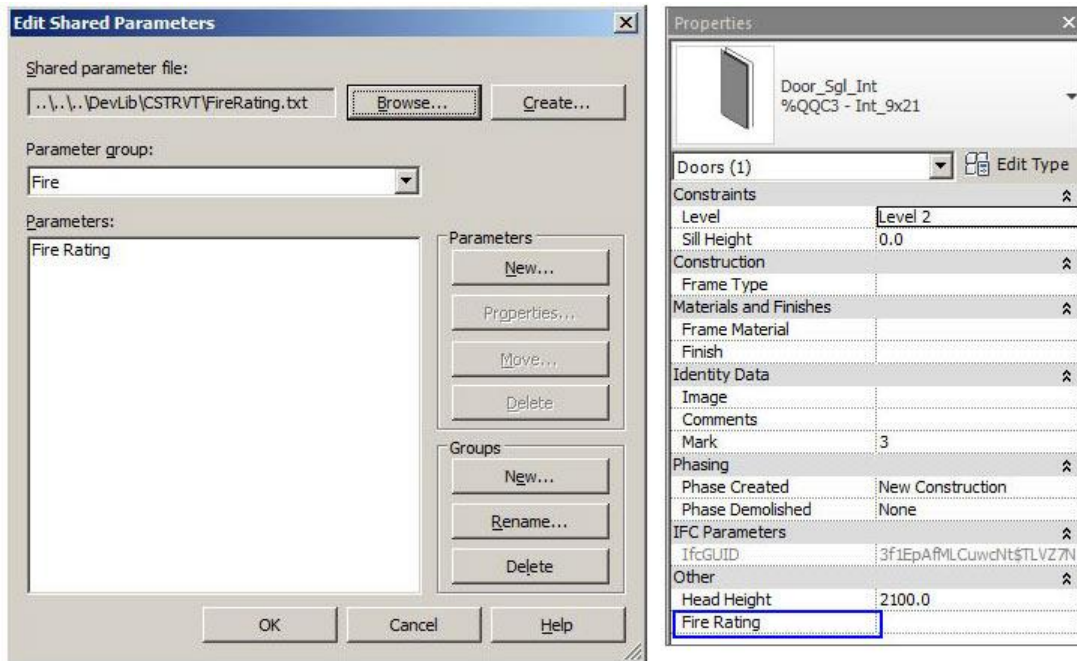


**Kontrol:**

Project Parameters:



Shared Parameters og Door Instance Parameters:



Følgende kan konstateres:

- *Shared Parameter* fil oprettet i CSTRVT mappen og forbundet med Revit.
- Parameter group oprettet: *Fire*
- Parameter oprettet: *Fire Rating*

Denne parameter er tilføjet dørenes egenskaber.

<Fire Rating er en *native* type parameter på døre i 2015, så det er måske lidt tvetydigt med en instance, men princippet fejler ikke noget>

## 1.3 EXPORT TO EXCEL

Add-Ins > Parameters > Fire Rating - Export to Excel

Der dukker ingen besked op i revit. Til gengæld åbner Excel med følgende ark:

	A	B	C	D	E	F	G
1	ID	Level	Tag	Fire Rating			
2	337697	Level 1	1	0			
3	350328	Level 1	2	0			
4	354400	Level 2	3	0			
5	354577	Level 2	4	0			
6	354622	Level 2	5	0			
7							
8							

Information er eksporteret succesfuldt!

Fire Rating står til 0 i arket. Det antyder, at parametertypen i revit er integer eller number, da værdier ikke var udfyldt. Man kan således ikke tilføje eksempelvis EI60 eller BD60, men lige nu gælder det re-eksport.

Forskellige værdier tilføjes i Revit. Yderdøre i stueplan: 60. Inderdøre på 1. sal: 30

Et dør-skema er oprettet for lettere kontrol.

ID er ikke tilgængeligt i schedules og Fire Rating er native Type Parameter, så begge felter må tilføjes:

<Checkup - Doors>			
A	B	C	D
Level	Mark	Fire Rating	Fire Rating
Level 1	1	60	
Level 1	2	60	
Level 2	3	30	
Level 2	4	30	
Level 2	5	30	

Add-Ins > Parameters > Fire Rating - Export to Excel

Succes. Værdierne eksporteres, prompt: bekræft overskriv, nyt ark.

FireRating.xls - Excel

FILER HJEM INDSÆT SIDELAYOUT FORMLER DATA GENNEM

Sæt ind

Udklipsholder Skrifttype Justering Tal

N16

	A	B	C	D	E	F	G
1	ID	Level	Tag	Fire Rating			
2	337697	Level 1	1	60			
3	350328	Level 1	2	60			
4	354400	Level 2	3	30			
5	354577	Level 2	4	30			
6	354622	Level 2	5	30			
7							
8							

Fire Rating

KLAR

Tid til næste skridt.

## 1.4 IMPORT FROM EXCEL

Først justeres værdierne en smule, så ændringer efterfølgende kan registreres i revit.

Værdierne importeres og er tilføjet dørene:

<Checkup - Doors>			
A	B	C	D
Level	Mark	Fire Rating	Fire Rating
Level 1	1	61	
Level 1	2	62	
Level 2	3	31	
Level 2	4	32	
Level 2	5	33	

Excel åbner endda med det pågældende ark, så man kan se hvad det er man har importeret.

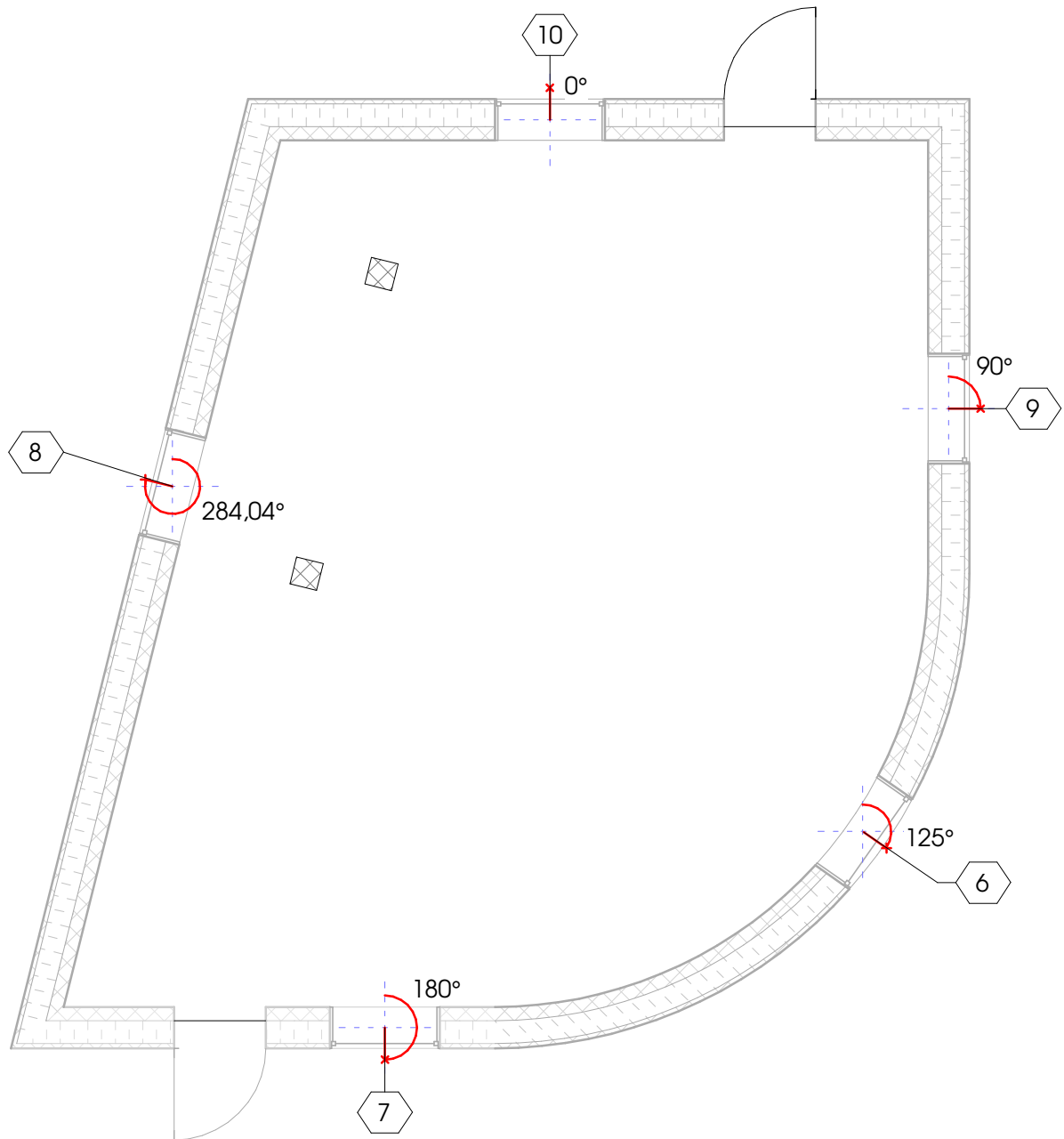


# APPENDIKS G

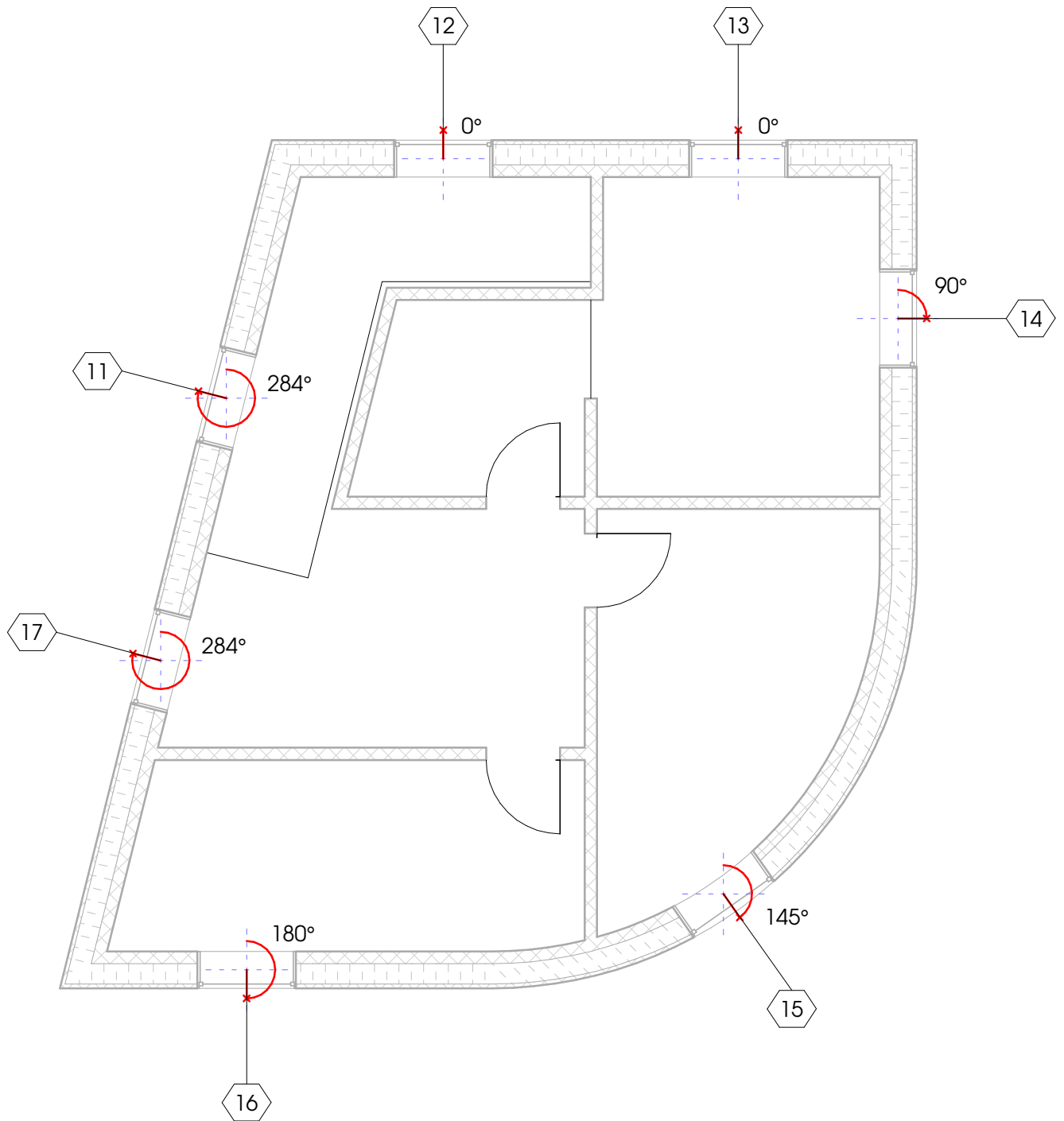
17 instances from link: tm\_ARK-S\_02.rvt

FamilyType	ID	Mark	Areal [m2]	Orienterin	Hældning	X:	Y:	Z:	
%QQA1 - 1200x1200	337156	6	1.44	125	90	0.819152	-0.57358	0	(0.819152044, -0.573576436, 0.000000000)
%QQA1 - 1200x1200	337798	7	1.44	180	90	-3.2E-15	-1	0	(0.000000000, -1.000000000, 0.000000000)
%QQA1 - 1200x1200	337828	8	1.44	284.04	90	-0.97014	0.242536	0	(-0.970142500, 0.242535625, 0.000000000)
%QQA1 - 1200x1200	337830	9	1.44	90	90	1	-1.6E-15	0	(1.000000000, 0.000000000, 0.000000000)
%QQA1 - 1200x1200	337898	10	1.44	0	90	3.23E-15	1	0	(0.000000000, 1.000000000, 0.000000000)
%QQA1 - 1200x1200	339318	11	1.44	284.04	90	-0.97014	0.242536	0	(-0.970142500, 0.242535625, 0.000000000)
%QQA1 - 1200x1200	339371	12	1.44	0	90	3.23E-15	1	0	(0.000000000, 1.000000000, 0.000000000)
%QQA1 - 1200x1200	339398	13	1.44	0	90	3.23E-15	1	0	(0.000000000, 1.000000000, 0.000000000)
%QQA1 - 1200x1200	339472	14	1.44	90	90	1	-1.6E-15	0	(1.000000000, 0.000000000, 0.000000000)
%QQA1 - 1200x1200	339516	15	1.44	145	90	0.573576	-0.81915	0	(0.573576436, -0.819152044, 0.000000000)
%QQA1 - 1200x1200	339530	16	1.44	180	90	-3.2E-15	-1	0	(0.000000000, -1.000000000, 0.000000000)
%QQA1 - 1200x1200	339556	17	1.44	284.04	90	-0.97014	0.242536	0	(-0.970142500, 0.242535625, 0.000000000)
%QQA3 - 0610x0700	410576	21	0.43	90	65	0.422618	-6.6E-16	-0.90631	(0.422618262, 0.000000000, -0.906307787)
%QQA3 - 0610x0700	410582	19	0.43	180	30	-2.8E-15	-0.86603	-0.5	(0.000000000, -0.866025404, -0.500000000)
%QQA3 - 0610x0700	410828	18	0.43	180	30	-2.8E-15	-0.86603	-0.5	(0.000000000, -0.866025404, -0.500000000)
%QQA3 - 0610x0700	410930	20	0.43	284.04	65	-0.41	0.1025	-0.90631	(-0.409999937, 0.102499984, -0.906307787)
%QQA3 - 0610x0700	411100	22	0.43	0	50	2.14E-15	0.642788	-0.76604	(0.000000000, 0.642787610, -0.766044443)

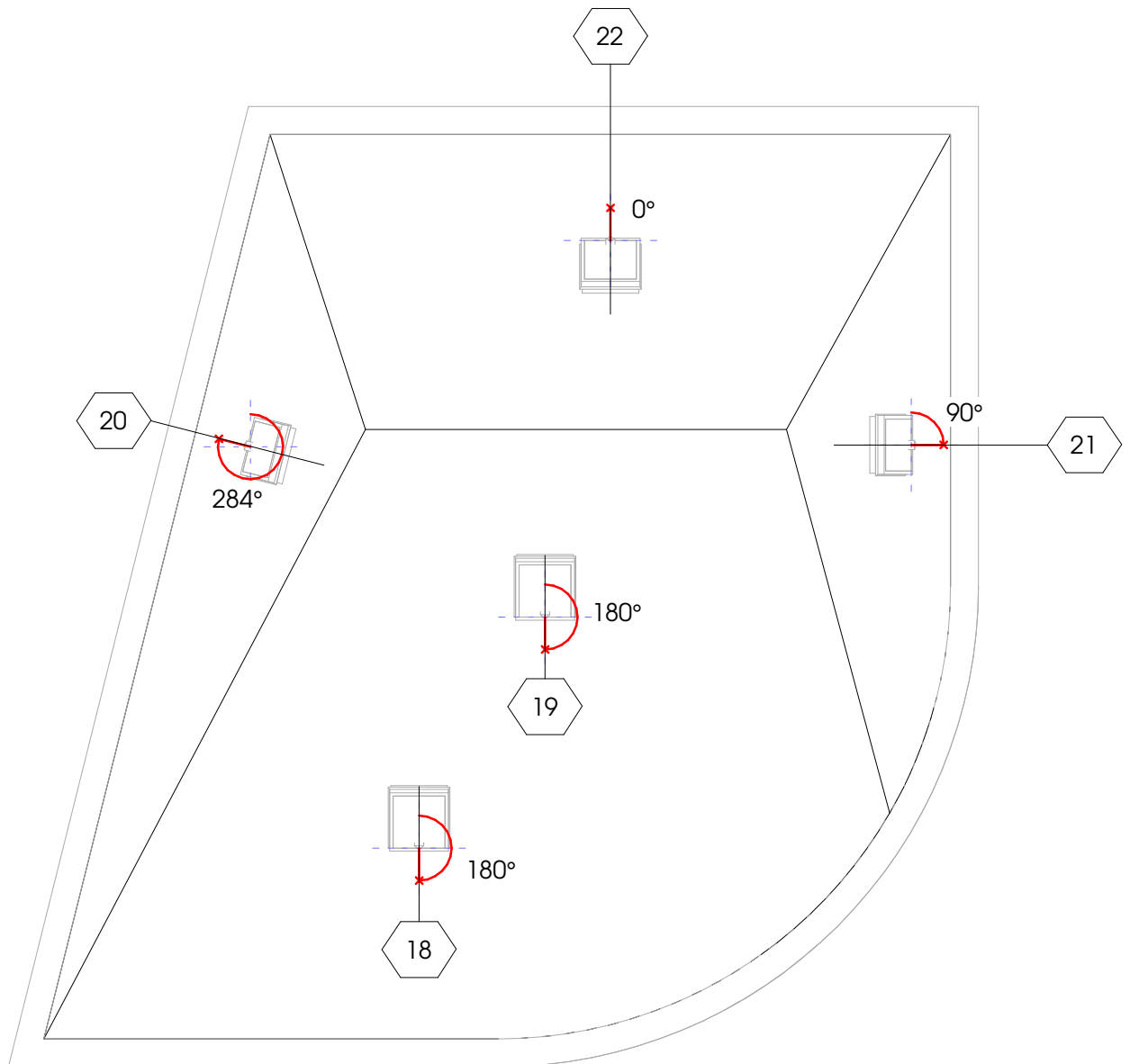
# APPENDIKS G

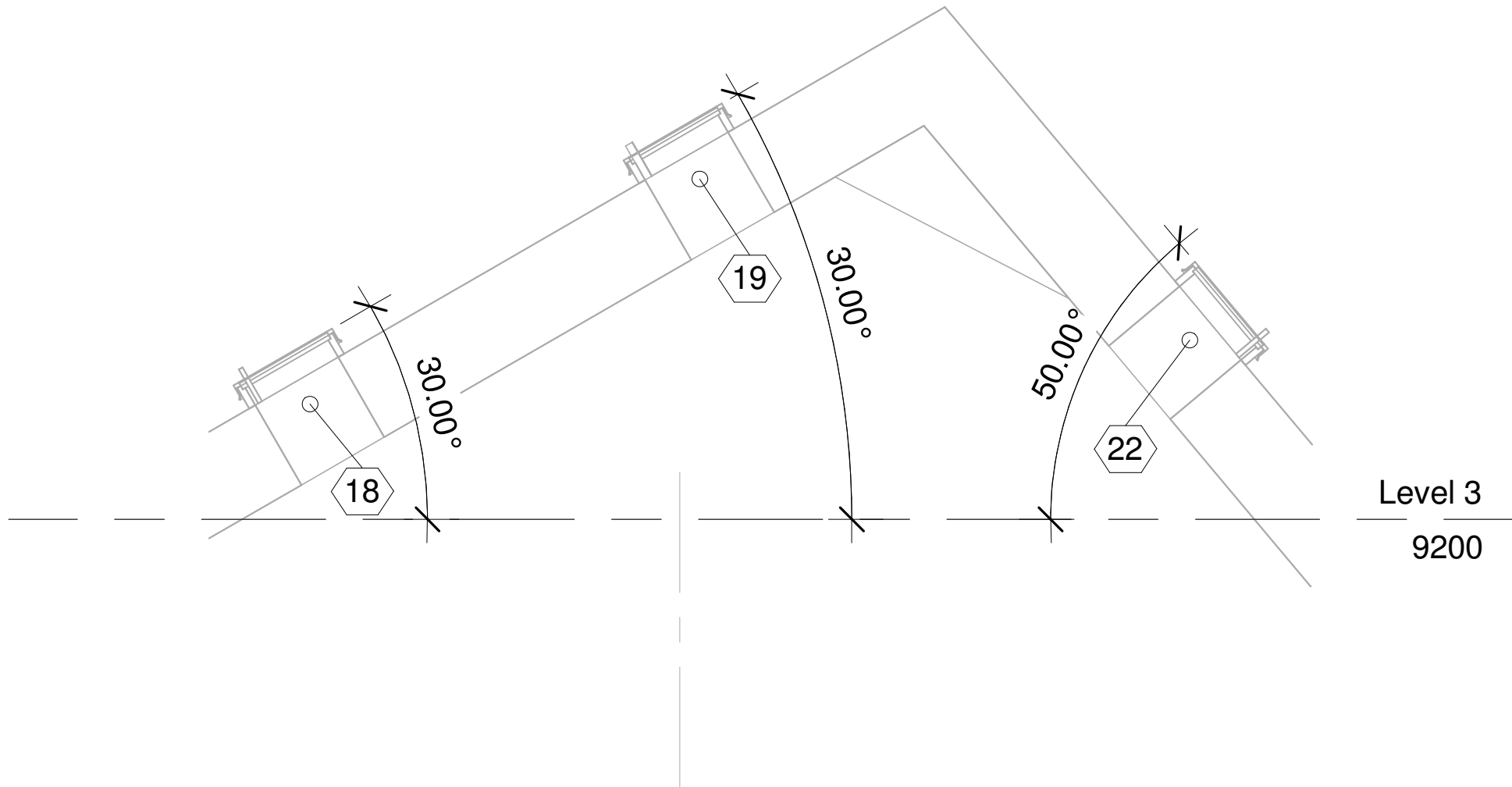


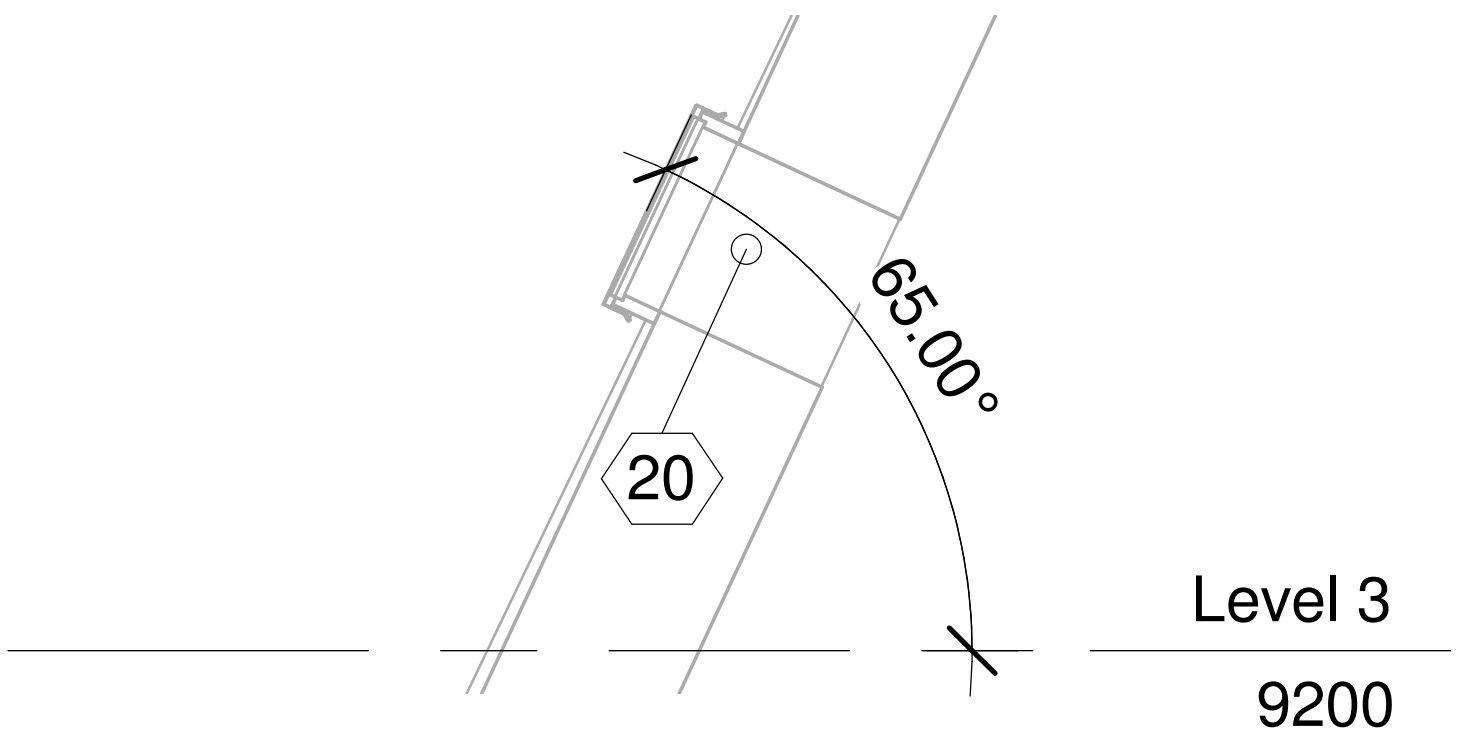
# APPENDIKS G

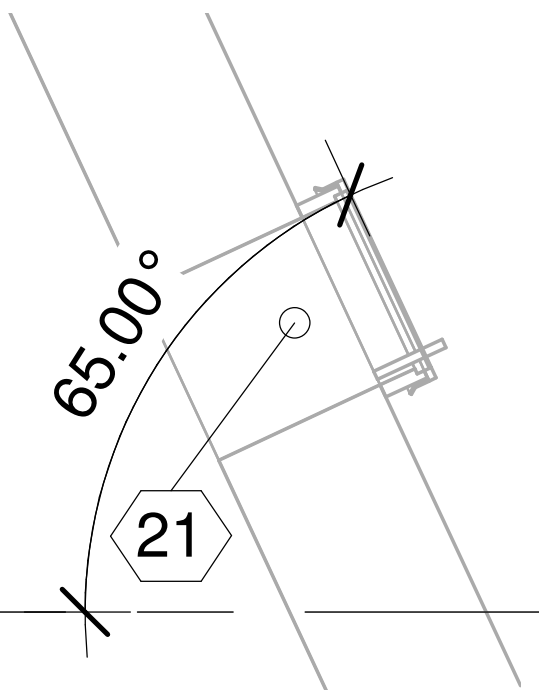


# APPENDIKS G









Level 3

9200