



AALBORG UNIVERSITET

INFORMATIONSINDHOLDET I BYGNINGSMODELLER

Gevinstpotentialerne ved kravspecifikation igennem Model View Definition

af

Nadim Saboor
Bygningsinformatik

Kandidatafhandling
Januar 2019

TITELBLAD

Rapport titel: Informationsindholdet i Bygningsmodeller
Undertitel: Gevinstpotentialerne ved kravspecifikation igennem Model View Definition

Uddannelsesinstitution Aalborg universitet, Institut for Byggeri og Anlæg
Uddannelseslinje: Bygningsinformatik
Semester: Afgangsprojekt
Projektperiode: september 2018 – Januar 2019
ECTS: 30

Vejleder: Kjeld Svidt
Samarbejdspartner: Mads Carlsen
Samarbejdsvirksomhed: Årstiderne Arkitekter

Oplag: Digital publicering
Udgivelsesdato: 10. januar 2019
Sprog: Dansk

Forfatter: Nadim Saboor

Underskrift:



Ved at underskrive dette dokument bekræfter jeg, at jeg selv har udarbejdet projektrapporten og at jeg selv hæfter for rapporten indhold.



AALBORG UNIVERSITET

FORORD

Denne kandidatafhandling er udarbejdet i forbindelse med uddannelsen Bygningsinformatik (cand. scient. techn.). Afhandlingen afslutter et toårigt forløb, fordelt over fire semestre ved fakultetet byggeri og anlæg i Aalborg Universitet.

Afhandlingen henvender sig til byggebranchen og bygger på eksisterende problemstillinger i forbindelse med validering af informationsindholdet i digitale bygningsmodeller. Disse belyses indledningsvis hvortil der udvikles et IT-baseret løsningsforslag.

I forbindelse med tilblivelsen af rapporten, ønsker jeg at takke Kjeld Svidt for akademisk vejledning, Mads Carlsen for hans motiverende lederskab og samarbejdsvirksomheden Årstiderne Arkitekter, som tog godt imod mig og fik mig til at føle som en del af teamet under projektskrivningsperioden. Yderligere, vil jeg gerne takke interviewrespondenterne for deres interviewdeltagelse, herunder IKT-koordinator i Årstiderne Arkitekter Frederik Bæk, samt formand for bygherreforening og afdelingschef for teknisk videntcenter Michael Ørsted.

Nadim S.

RESUME

Mål for undersøgelsen

Kvalitative undersøgelser påpeger, at den nuværende kvalitetssikring af bygningsmodeller er både tidskrævende og fejlbehæftet. Nærværende undersøgelse vurderer valideringsfunktionen i Ifc-Doc i forhold til effektivisering af valideringsprocessen.

Metode

Rapporten følger datamodelleringsprocessen til at definere en delmængde i Ifc-skemaet i henhold til en række egenskabsdata (LOD-dk-niveauer), som fremgår i bygningsdelsspecifikationer udviklet af DiKon og BIM7aa. Denne delmængde kaldes for en Model View Definition (MVD). MVD-udviklingen danner grundlaget for en vurdering af valideringsfunktionen i IfcDoc.

Resultater

MVD-udvikling på baggrund af informationsniveauerne effektiviserer valideringsprocessen igennem brugen af en dedikeret funktion i IfcDoc. En af fordelene ved MVD-tilgangen, er at man kan udnytte objektorienterede datamodelleringsprincipper, herunder generaliseringsevnen. Det modvirker gentagelse og dobbelt arbejde. Dog er det ikke kravstillelse på attributniveau, som er kerneformålet ved MVD-tilgangen.

Konklusion

Løsningsforslaget udlover effektiviseringer i forhold til valideringsprocessen. Dog adresseres problemet ikke ved roden. Attributterne findes forskellige steder og hedder noget forskelligt alt afhængig af softwaren. MVD-tilgangen bør i stedet anvendes som en neutral kravspecifikationsform baseret på den samlede informationsbehov ved enkelte aktører brugsformål. MVD-konceptet er ikke ment til validering af informationsniveauer på et attributniveau.

ABSTRACT

Objective

Qualitative studies point out that the current quality assurance of building models is both time consuming and defective. This study evaluates the integrated validation function in IfcDoc in relation to efficiency of the validation process.

Method

The report follows the data modeling process to define a subset in the Ifc-schema according to a set of property data (LOD-dk levels), which are stated in “byggningsdelsspecifikationer” developed by DiKon and BIM7aa. This subset is called a Model View Definition (MVD). The MVD development forms the basis for an assessment of the validation function in IfcDoc.

Result

MVD development based on the information levels streamlines the validation process through the use of a dedicated function in IfcDoc. One of the benefits of the MVD approach enables exploitation of object-oriented data modeling principles including the generalization. It counteracts repetition and duplication. However, the core objective of the MVD approach is not validation at attribute level.

Conclusion

The developed solution has potentials to streamlining the cumbersome validations process. However, the problem is not addressed by the root. The MVD approach should instead be used as a neutral requirement specification form based on the total information needs of some actors. And not used to validate information levels at an attribute level, since the attributes are somewhat different and are located in different locations depending on the software.

INDHOLD:

I. BESKRIVENDE	6
1.1 INTRODUKTION: INDHOLDET I BYGNINGSMODELLER HAR FÅET EN CENTRAL ROLLE	7
1.2 STATE OF THE ART: LEVEL OF DETAIL (LOD)	8
1.3 PROBLEMSTILLING: TIDSKRÆVENDE VALIDERINGSPROCES	13
1.4 MODEL VIEW DEFINITION	15
HVORDAN KAN UDVIKLING AF MODEL VIEW DEFINITION PÅ BAGGRUND AF BYGNINGSDELSSPECIFIKATIONER EFFEKTIVISERE VALIDERINGSPROCESSEN?	16
1.5 METODE: DATAMODELLERINGSPROCESSEN	17
1.6 LITTERATURSTUDIE	18
1.7 DATAINDSAMLING	19
1.8 CASE: LINDBJERGSKOLEN (LBS)	21
1.9 AFGRÆNSNING	22
II. ANALYSERENDE	23
2.1 VIRKELIGHED	24
DELKONKLUSION	31
2.2 KONCEPTUALISERING: UNIFIED MODELLING LANGUAGE (UML)	32
DELKONKLUSION: KONCEPTUALISERING I UML	40
2.3 REALISERING: UDVIKLING AF MVD BASERET PÅ KONCEPTUEL MODEL	42
KONKLUSION PÅ ANALYSERENDE DEL	49
III. VURDERENDE	50
3.1 VALIDERING AF EGENSKABSDATA TYPE A, B OG C	51
3.2 DISKUSSION & RESULTATER	53
IV. KONKLUSION OG PERSPEKTIVERING	62
V. BIBLIOGRAPHY	65
VI. FIGURLISTE	66
VIII. BILAGOVERSIGT	67

I. BESKRIVENDE

Rapporten er inddelt i tre dele. En beskrivende, en analyserende og en vurderende del. Nærværende kapitel udgør den første del i rapporten og har til formål at beskrive undersøgelsesområdet. Kapitlet beskriver emne, problemformulering og metode igennem 9 overskrifter.

Emne: Informationsindholdet i bygningsmodeller

Problem: Tidskrævende valideringsproces i Solibri

Undersøgelsesformål: At udvikle Model View Definition på baggrund af informationsniveauer til vurdering af valideringsfunktionen i IfcDoc

Metode: Rapporten følger datamodelleringsprocessen til at definere en delmængde i Ifc-skemaet i henhold til en række egenskabsdata (LOD-dk-niveaur), som fremgår i bygningsdelsspecifikationer udviklet af DiKon og BIM7aa.

1.1 Introduktion: Indholdet i bygningsmodeller har fået en central rolle

Informationsteknologi (IT) er computerbaseret teknologi som omfatter udvikling, behandling, lagring og udbredelse af data. (ordnet, 2018) I byggeindustrien bruges IT oftest i forbindelse med kommunikation af information. Af denne årsag, anvendes den mere konkrete betegnelse, informations- og kommunikationsteknologi (IKT). (ordnet.dk)

I 2003 offentliggjorde regeringen en ny byggepolitisk handlingsplan ”Staten som bygherre”. Planen blev lanceret på baggrund af en konstatering om, at produktiviteten i dansk byggeri i en årrække havde udviklet sig langsommere end landets øvrige erhverv. Regeringen igangsatte flere initiativer for at modvirke den lave produktivitet i branchen, heriblandt initiativet ”Det Digitale Byggeri”. Strategien bag Det Digitale Byggeri var at udvikle et fundament af IT-standarder og fælles retningslinjer for branchen, og at implementere disse i branchen ved at stille dem som krav i forbindelse med statslige byggeprojekter. (Danmarks Regering, 2003)

Det Digitale Byggeri beskriver en digital byggeproces, hvor data akkumuleres, processeres, kopieres, genbruges og udtrækkes til et givent formål. (Jespersen, 2008) Principperne for Det digitale Byggeri bygger på Building Information Modelling (BIM). BIM er en digital udgave af fysiske og funktionelle karakterer af en bygning. En BIM-model kan opfattes som et delt videntcenter, som indeholder informationer om et byggeprojekt, hvor aktørerne i kraft heraf kan fortage intelligente beslutninger igennem hele bygningslivscyklus fra ide til nedrivning. BIM forudsætter samarbejde på tværs af aktørerne i de forskellige byggefaser, sådan så alle aktører som minimum kan tilføje, udtrække, opdatere og ændre informationer i BIM-modellen, sådan så modellen understøtter og reflekterer det specifikke behov hos den pågældende aktør (Eastman *et al.*, 2011)

BIM-konceptet stammer fra 70’erne og bygger på problemstillinger ved konventionelle CAD-producerede tegninger - hvor man optegnede linjer, som repræsenterede objekter i virkeligheden. Computere havde svært ved at indlæse og fortolke tegningerne, og forhindrede dermed automatisering. Principperne for BIM bygger således på at effektivisere at forstærke samarbejdet i blandt byggeprojekters parter ved at effektivisere informationsudveksling og reducere ressourcespild i kraft af bedre udnyttelse af IKT-værktøjer.

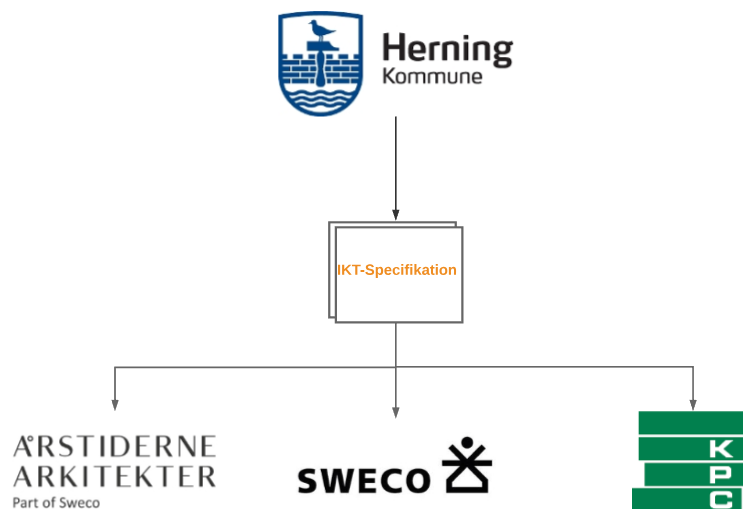
Informationsindholdet i bygningsmodeller er blevet en væsentlig prioritering byggesagens aktører. Der er ikke en universel definition for hvad BIM-modeller skal indeholde. Indholdet er oftest afhængigt af et givent brugsformål. Formålet for brug af modellen afgør hvilke informationer skal være tilstede. Derfor er det væsentligt, at den modtagne part kommunikerer og specificerer et givent brugsformål. Jo konkret og jo bedre denne informationsefterspørgsel defineres, desto bedre bliver grundlaget for en vellykket leverance. (Kreider, 2010)

Kravspecifikation i forhold til indholdet i bygningsmodeller har fået en central rolle, når det omhandler effektiviseret brug af IKT-værktøjer. Rapporten tager således afsat i en beskrivelse af hvordan kravspecifikation fortages med udgangspunkt i indsamlede data.

1.2 State of the art: Level of Detail (LOD)

Ved de konventionelle designpraksis, brugte man tegningsskalaer, som reference til behovet for geometrisk detaljeringsgrad i de forskellige faser og brugsformål. Level of Development (LOD) er et analog-koncept. Denne beskriver informationsindholdet i BIM-modeller igennem standardiserede detaljeringsgrader. LOD-niveauer referer til bygningsmodellers informationsniveau. (BIMForum, 2018) I dette afsnit belyses kravspecifikation i forhold til informationsindholdet i bygningsmodeller med udgangspunkt i LBS-casen (jf. afsnit 1.8).

Kravspecifikation fra bygherre til øvrige



Figur 1 Bygherren stiller krav til øvrige parter igennem IKT-specifikationer (egen tilvirkning)

Herning Kommune var både driftsherren og bygherren på sagen. Herning Kommune anvendte IKT-specifikation på sagen¹ (jf. bilag: IKT-specifikation). Denne beskriver informationskravet til bygningsmodeller ved reference til informationsniveauer². Nedenstående tabel danner oversigt over bygherrens krav til informationsindholdet i bygningsmodeller.

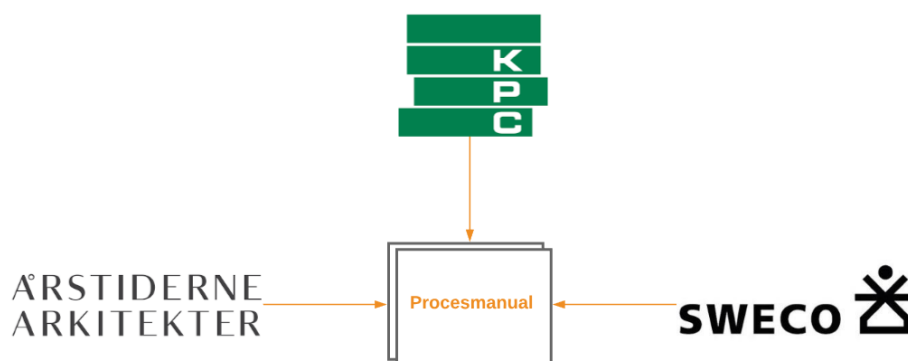
¹ IKT-specifikation gør det enkelt at definere omfanget af IKT-ydelserne (bips.dk, 2018)

² CCS Informationsniveauer er en del af cuneco classification system (CCS). CCS giver bygge- og anlægsbranchen et fælles sprog og metoder til at skabe entydig udveksling af informationer gennem hele byggeprocessen fra idé til drift. (bips, 2014)

	Informationsniveau	Anvendelsesformål
Programfase	0	
Disposition- og projektforslagsfase	1-2	Konsistens- og kollisionskontrol
Forprojektfase	1-2	Konsistens- og kollisionskontrol BSim simulering
Hovedprojektfase	3-4	Konsistens- og kollisionskontrol
Udførelsesfase	5	Konsistens- og kollisionskontrol BSim simulering
Brugs- og driftsfase	6	BSim simulering

Tabel 1 Kravsstillelse til informationsindholdet i fagmodeller

På baggrund af IKT-specifikationer udarbejdede parterne en IKT-procesmanual til at afstemme hvordan man i fællesskab, og på mest effektive måde, når frem til aftalte leverancer. Procesmanualen kan opfattes som en konkretisering af Herning kommunes IKT-ydelsesspecifikationer.



Figur 2 Aftalegrundlaget blandt rådgivende og udførende (egen tilvirkning)

Procesmanualen definerer og fordeler ansvarsområder i forbindelse med BIM-projekteringen.

I procesmanualen er der redegjort for fastlæggelse af geometri, samt modelleringsmæssige forhold, som kan have indflydelse på udveksling og/eller eksport og bygningsmodellerne ved

reference til en foruddefineret standard med udgangspunkt i publikationen ”Leverance specifikation for bygningsmodeller” udarbejdet af Digital Konvergens (DiKon)³. I procesmanual indgår det, at rådgiverne skal udarbejde leverancespecifikation for bygningsmodeller for fastlæggelse af fases minimumskrav, som er beskrevet i IKT-specifikationerne (jf. bilag procesmanual). Leverancespecifikationer blev således brugt internt hos rådgivende, til at leve op til bygherrens kravsspecifikation.

Kravspecifikation internt: leverancespecifikation & bygningsdelsspecifikation

Konceptet bag leverancespecifikationer er udformet på baggrund af en fælles anerkendelse af den væsentlige rolle som bygningsdele (objekter) og tilhørende egenskabsdata (informationer) har fået for byggeriets parter.

”Leverancespecifikationer dækker behovet for entydigt at beskrive indholdet af en bygningsmodel i forhold til bygningsdelens pålidelighed, geometriske præstation og tilhørende egenskabsdata.” (dikon.info, 2018)

Leverancespecifikationer er en template i Excel, som udfyldes projektspecifikt forud for projekteringen. Templatet er en platform for dialog med øvrige parter, der tilgår projekteringen. I kraft af leverancespecifikationer, kan parterne afklare ansvarsområder i forhold til informationsindholdet i enkelte bygningsdele inden projekteringen igangsættes.

Kategori Category	Fagområde Discipline	Bygningsdel Building Part	Bygningsdel Building Part	Bygningsdels- specifikationer (Building Part Specification)	LOD DK LOR LOG LOI Dispositions- forslag Outline proposal
Rum (Spaces)					
	K01 - Arkitekt (Arc)	Rum	Room	Rum	200
Areal (Areas)					
	K01 - Arkitekt (Arc)	Areal	Area	Rum	200
Fundament (Foundation)					
	K09 - Konstruktioner (Str)	Punktfundament	Point foundation	Fundament	
	K09 - Konstruktioner (Str)	Stribefundament	Strip foundation	Fundament	
	K09 - Konstruktioner (Str)	Pladefundament	Plate foundation	Fundament	
	K09 - Konstruktioner (Str)	Terrændæk	Ground floor slab	Betondæk	
	K09 - Konstruktioner (Str)	Grovbeton	Mass concrete foundation	Fundament	
Væg (Wall)					
	K01 - Arkitekt (Arc)	Bærende væg	Load bearing wall	Væg	200
	K01 - Arkitekt (Arc)	Let væg	Partition wall	Væg	200
	K09 - Konstruktioner (Str)	Betonvæg, præfab	Concrete wall, prefab	Betonvæg	
	K09 - Konstruktioner (Str)	tonvæg, pladsstøbt	Concrete wall, in situ	Betonvæg	

Figur 3 Udklip af leverancespecifikationer på casen (jf. bilag: leverancespecifikation)

Ovenstående figur er et udsnit af en udfyldt leverancespecifikation på LBS-casen. Det fremgår af figuren, at arkitekten er ansvarlig for modulering af rum, vægge og fundamenter svarende til en Level of Development (LOD) på 200 i dispositionsforslaget. Ansvarstildeling i forhold til informationsindholdet i enkelte bygningsdele forekommer ved reference til LOD-

³ DiKon udspringer af et fælles ønske om effektivisering af den digitale byggeproces blandt syv samarbejdspartnere (dikon.info/om-dikon/)

niveauer. I forlængelse af leverancespecifikationer har DiKon i samarbejde med BIM7AA⁴ udarbejdet bygningsdelsspecifikationer.

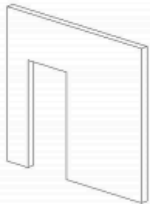
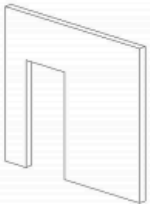
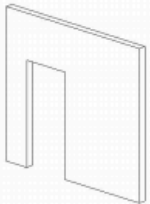
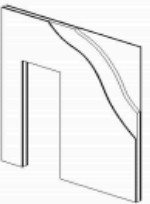
Bygningsdelsspecifikationer

Bygningsdelsspecifikationer er udviklet for at etablere en simpel metode til at beskrive indholdet i bygningsmodellen på et givent tidspunkt. LOD-Terminologien anvendt i bygningsdelsspecifikation-publikationer har koblinger til øvrige LOD standarder og publikationer, herunder CCS-informationsniveauer. For at undgå sammenblanding med andre internationale LOD specifikationer, anvendes DK som en del af navngivningen for de danske niveauer –eksempelvis LOD 200 DK. (BIM7AA & DiKon, 2018)

LOD DK	LOD 200 DK	LOD 300 DK	LOD 325 DK	LOD 400 DK
CCS Informationsniveauer	Info.niv. 3	Info.niv. 4	Info.niv. 5	Info.niv. 6
BIPS C102 Informationsniveauer	Info.niv. 2	Info.niv. 3	Info.niv. 4	Info.niv. 5
YBL 2018, Digital Projektering	Forventet geometri	Fastlagt geometri	Endelig geometri	-

Figur 4 LOD-DK-kobling til øvrige standarder (BIM7AA & DiKon, 2018)

Sammenhængen mellem bygningsdelsspecifikationer og kravspecifikation i forhold til indholdet i bygningsmodeller, beskrives ud fra nedenstående figur.

LOD 200 DK Informationsniveau 3	LOD 300 DK Informationsniveau 4	LOD 325 DK Informationsniveau 5	LOD 400 DK Informationsniveau 6
LOR 200	LOR 300	LOR 325	LOR 400
FORVENTET Vægge defineres på forventet niveau for geometri, placering og tilhørende egenskabsdata.	FASTLAGT Vægge defineres på fastlagt niveau for geometri, placering og tilhørende egenskabsdata.	ENDELIG Vægge defineres på endeligt niveau for geometri, placering og tilhørende egenskabsdata.	ENDELIG DETALJERET Vægge defineres på endelig detaljeret niveau for geometri, placering og tilhørende egenskabsdata i henhold til faktiske produktvalg.
LOG 200	LOG 300	LOG 325	LOG 400
GENERISK NIVEAU  Vægge, inkl. større åbninger, modelleres i maks. ydre kontur opdelt på overordnede typer.	TYPE-NIVEAU  Vægge, inkl. større åbninger, modelleres i maks. ydre kontur opdelt på typer.	DETALJERET TYPE-NIVEAU  Vægge, inkl. større åbninger, modelleres i maks. ydre kontur opdelt på typer.	PRODUKTIONS-NIVEAU  Vægge, inkl. større åbninger, modelleres med konstruktionslag opdelt på typer. Sekundære konstruktionslag kan sammenlægges. Større huller og komponenter modelleres.
LOI 200	LOI 300	LOI 325	LOI 400
TILHØRENDE EGENSKABSDATA Typenavn Tykkelse	TILHØRENDE EGENSKABSDATA Typenavn Tykkelse	TILHØRENDE EGENSKABSDATA Typenavn Tykkelse Placering: Etage Konstruktionsopbygning	TILHØRENDE EGENSKABSDATA Typenavn Tykkelse Placering: Etage Konstruktionsopbygning

⁴ BIM7AA er et samarbejde mellem 7 arkitektvirksomheder til formål at optimere BIM-baseret samarbejde i fællesskab (bim7aa.dk)

Figur 5 Bygningsdelsspecifikation over vægge (DiKon & BIM7AA, 2018)

Bygningsdelsspecifikationer fordeler enkelte bygningsdele over fire LOD-niveauer. LOD 200 DK, som er det laveste niveau, stiller krav til modellering af mindst to obligatoriske egenskabsdata, herunder typenavn og tykkelse. LOD 400 DK, som er det højeste niveau, kræver mindst 4 obligatoriske egenskabsdata. For væggen gælder følgende egenskabsdata: typenavn, tykkelse, placering og konstruktionsopbygning.

Når det for eksempel aftales i leverancespecifikationen, at arkitekten har til ansvar at modulere vægge i henhold til LOD DK 200 ved udgangen af dispositionsforslaget, kravstilles der til, at vægge moduleres hvori parametrene typenavn og tykkelse indgår. Ligeledes gælder det for øvrige bygningsdele og LOD-DK-niveauer.

Bygningsdelsspecifikationer standardiserer detaljeringsgraden og sikrer en ensartet forståelse af over hvad enkelte LOD-niveauer rummer. Leverancespecifikationer specificerer ansvarsroller i forhold til modulering af enkelte bygningsdele. Samlet har de til formål at effektivisere informationsudvekslingen blandt parterne i byggesager.

1.3 Problemstilling: Tidskrævende Valideringsproces

Behovet for validering af indholdet i bygningsmodeller opstår når informationer skifter hænder. Valideringsprocessen har til formål at kvalitetssikre informationsindholdet i bygningsdele i henhold aftalegrundlaget. På nuværende tidspunkt forgår validering igennem modelgranskning.

Modelgranskning forgår ved, at afsenderen eksporterer en bygningsmodel i IFC-format, hvorefter den modtagne part åbner modellen i Solibri Model Checker. Solibri har en regelsætsfunktion, som bruges til automatisering af modelgranskning. I forbindelse med regelsættene, oplevede man mange udfordringer på LBS-sagen. Disse belyses i interview med Frederik Bæk, IKT-koordinator på LBS-sagen.

”Vi opdagede, at de regelsæt vi opsætter i Solibri ikke opdager alle typer fejl. Der var en fejl eller sådan noget i regelsættet, som gjorde, at vi blev nød til at lave manuel kontrol af samtlige bjælker. Det tog meget tid.”

”Vi oplever ofte, at Solibri melder tusinder af fejl bare indenfor det samme regelsæt. Problematikken ligger i, at man er nødt til at gennemgå dem alle sammen manuelt. Og sortere dem fra.”

Frederik Bæk, Årstiderne Arkitekter

Teknisk Videntcenter, KBH-lufthavn

For at undersøge, om problemet med modelgranskning i Solibri eksisterer på et højere plan, blev der foretaget interview med Michael Ørsted, formand for bygherreforening samt afdelingsleder i teknisk videntcenter i Københavns lufthavn. Afdelingen håndterer alt teknisk datainformation tilhørende københavns lufthavn, inklusiv BIM-modeller fra bygge- og anlæg projekter. Funktionerne består af kravspecificering, kvalitetssikring samt udlevering af data. Michael Ørsted beskriver kravspecifikationsformen, som følgende:

”Vi har både krav til informationsniveau, så har vi også yderligere krav til hvordan man skal modulere og aflevere. Ved siden af det, så har vi også nogle krav som kommer fra vores drift, som kommer i form af projektmanueller. I de her står for eksempel noget med at et eller andet materiale skal være at galvaniseret stål.”

Kravspecificering forgår igennem blandt andet CCS-informationsniveauer og valideringen foregår igennem modelgranskning i Solibri. Afdelingen samler informationer og ligger dem i ”de rigtige hylder”, så de kan genbruges til et givent anvendelsesformål i efterfølgende projekter. Afdelingen har derfor meget fokus på kvalitetssikring af data. Kravspecificering og valideringsprocessen i afdelingen beskrives som følgende:

”Lige nu sker det ved, at vi bruger nogle regelsæt i Solibri til at sikre at attributterne er til stede.”

Respondenten udtrykker, at afdelingen bruger alt for lang tid på kvalitetssikring af indleverede bygningsmodeller for at sikre, at de er moduleret i henhold til det aftalte

informationsniveau. Michael Ørsted er dog enig i, at en konkretisering af informationsefterspørgselen kunne gavne.

”Jeg er meget fortalende for, at man bliver så konkret som overhovedet muligt i IKT-specifikationer. Hvis vi kan blive helt skarpe omkring vores informationsbehov, så fjerner vi alle usikkerheder og vi kan få bedre pris og klima under projektopbygningen.”

På baggrund af ovenstående vurderes det, at kvalitetssikring ved model granskning i Solibri er både tidskrævende og fejlbehæftet. Nærværende undersøgelse, har således til formål at finde en afhjælpende løsning på den tidskrævende model granskning.

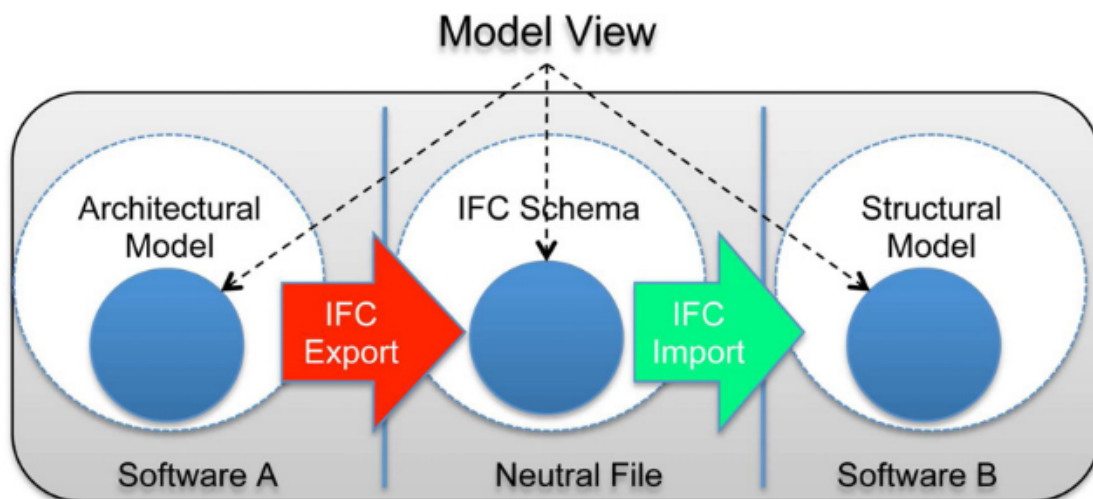
1.4 Model View Definition

Industry Foundation Class (IFC) er en neutral datamodel⁵, som er anerkendt for at understøtte dataudveksling på tværs af BIM-certificerede værktøjer. IFC-datamodellen understøtter, at BIM-værktøjer kan kommunikere og udveksle data mellem hinanden på trods af deres forskelligheder. Det kaldes for interoperabilitet. (Eastman et. al., 2011)

Som fælles udvekslingsformat kræver det, at IFC-datamodellen skal understøtte samtlige informationsudvekslinger igennem hele byggelivscyklussen. Datamodellens indhold er derfor overflødig og kræver specificering i forhold til et givent formål. På baggrund af det, har buildingSMART udviklet konceptet Model View Definition (MVD). Denne defineres som følgende:

“A subset of building product model schema that provides a complete representation of the information concepts needed for a particular information exchange in an AEC workflow” (Eastman et al., 2011)

MVD-konceptet betyder, at man kan specificere en delmængde i IFC-datamodellen baseret på et konkret informationsbehov. Det teoretiske grundlag for MVD og dens potentiale til at understøtte kravspecifikation kan beskrives igennem nedenstående figur.



Figur 6 Model View Definition (Venugopal et. Al., 2012)

Figuren viser et eksport-import-scenarie mellem en arkitektmodel og en ingeniørmodel. Det hvide cirkel i de enkelte båser, illustrere det samlede datamængde. Det blå cirkel i arkitektmodellen udgør en delmængde af data, som er nødvendigt for konstruktionsingeniøren. MVD-konceptet går ud på at definere en datamængde i IFC-datamodel, som adressere informationsbehovet ved et givent anvendelsesformål. (Venugopal et al., 2012)

⁵ En datamodel danner det underliggende struktur af en database, og indeholder en samling af formler og regler hvorigennem dataenheder, data relationer, og semantiske betydninger beskrives. (silberschatz, korth, 2002)

IfcDoc⁶ værktøjet kan bruges til udvikling af MVD. Værktøjet har en indbygget funktion, som automatisk validerer et Ifc-eksport i henhold til et defineret MVD. Denne rapport undersøger om den indbyggede valideringsfunktion i IfcDoc, kan effektivisere valideringsprocessen. De efterfølgende analyser og undersøgelser er således afgrænset med henblik på at konkludere på følgende problemformulering:

Hvordan kan udvikling af Model View Definition på baggrund af bygningsdelsspecifikationer effektivisere valideringsprocessen?

Næste afsnit beskriver blandt andet metoden for MVD-udvikling. Resten af rapporten er opdelt som følgende:

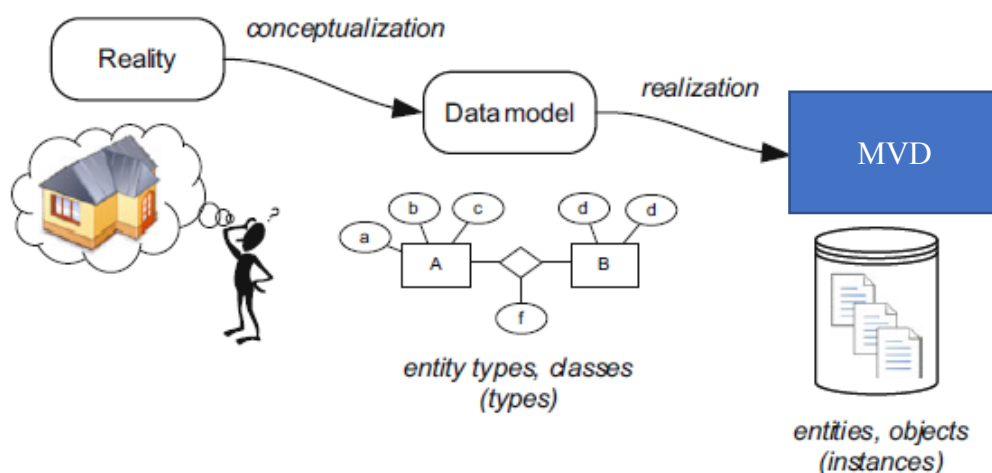
- Kapitel II: Følger datamodelleringsprocessen til MVD-udvikling på baggrund af bygningsdelsspecifikationer
- Kapitel III: Afprøver og vurderer valideringsfunktionen i Ifc-Doc
- Kapitel IV: Konklusion og perspektivering

⁶ buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool

1.5 Metode: Datamodelleringsprocessen

Model View Definition (MVD) er et lag eller delmængde i Ifc-skemaet, som adresserer et konkret informationsbehov i henhold til et givent anvendelsesformål. Ifc-skemaet er defineret i EXPRESS, som er et dedikeret sprog til udvikling af objektorienteret datamodeller. (Schenkt & Wilson, 1993)

Rapportens metode til udvikling af MVD er inspireret af datamodelleringsprocessen. Denne er opdelt i to overordnet trin: konceptualisering og realisering. I det første trin, abstraheres en virkelighed til en konceptuel datamodel, som beskriver virkeligheden ud fra Classes, attributter og relationer. (Booch, 2005)



Figur 7 datamodelleringsprocessen (Bormann et. al. 2018)

LOD-DK-informationsniveauer udgør realiteten i nærværende undersøgelse. LOD-niveauerne er specificeret i bygningsdelsspecifikationer. Denne konkretiseres, herefter abstraheres det igennem en passende konceptuel model for derefter at skabe grundlaget for en realisering i IFC-skemaet igennem MVD-udvikling.

1.6 Litteraturstudie

Afsnittet danner overblik over litteraturstudiet omkring emnet: Validering af bygningsmodeller. Afsnittet beskriver centrale fund i enkelte studier, som har præget og udformet rapportens metode til løsning af problemformuleringen.

Pinheiro et. al. udarbejdede et forslag til standardiseret metode til informationsudveksling mellem BIM og Building Energy Performance Simulation (BEPS) værktøj. Studiet foreslog anvendelse af Information Delivery Manual (IDM) og Model View Definition (MVD) til at specificer IFC-datamodellen iht. informationsbehovet hos BEPS-værktøjet. (Pinheiro *et al.*, 2018)

IDM rummer Exchange Requirements (ER), som oftest udvikles på papirbaserede formater. ER specificerer informationsbehovet ved en konkret udveksling og danner grundlaget for udvikling af MVD-koncepter. Informationerne i ER har ikke en tilstrækkelig logisk sammenhæng og beskriver ikke Relationer, Entities eller Attributes. Denne uklarhed kan resultere i fejl eller mangler i MVD-konceptet. Solihin et. al. foreslår en mere logisk sammenhængende beskrivelse af ER, som bygger på ontologiske principper. Og efterviser hvordan man i kraft af protege kan udarbejde ontologi med udgangspunkt i et ER, hvorefter denne eksporteres i XML format og importeres som MVD-koncept i IfcDoc, hvorefter der foretages en valideringsproces. (Lee, Eastman and Solihin, 2016)

Udvikling af MVD kræver principielle discipliner, som hviler på object-orienterede modulerings mekanismer. Venugopal et. al. konkluderede, at begrænsning af IFC-skemaet i henhold til et MVD-koncept behøver formelle definitioner mellem Entities, Attributes, og Relationer. (Venugopal *et al.*, 2012)

Fahad & Bus præsenterer og sammenstiller teknologierne: IfcDoc og Semantic Web Rule Language (SWRL). De vurderer teknologierne i forhold til validering af IFC-modeller og konkludere, at selvom begge teknologier kan bruges til formålet er SWRL mere fleksible og hurtigere, da denne ikke behøver et defineret MVD. (Fahad *et al.*, 2017)

Koch & König ligger vægt på vigtigheden af semantisk data, som inkluderer informationer vedrørende udførelsesmetoder, materialer og rumfunktioner. De præsenterer måder hvorpå semantisk-information kan blive struktureret igennem datamodellerings koncepter og notationer, herunder anvendelse af Unified Modeling Language (UML) til grafisk beskrivelse af Object-Oriented Modeller. (König, 2018)

Boch et. al. opdeler processen for datamodellering i to overordnet dele: konceptualisering og realisering. I den første del abstraheres realiteten til Entities, Attributes og Relations hvor disse beskrives igennem konceptuelle modeller. I den anden del realiseres data på baggrund af konceptualiseringen. (Booch, 2005)

1.7 Dataindsamling

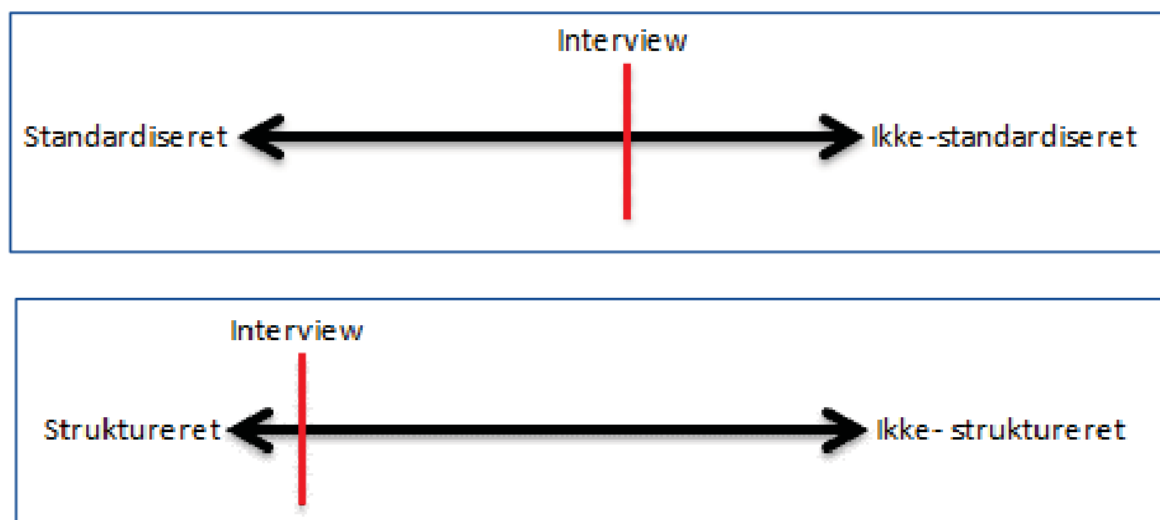
I forbindelse med specialet er der indsamlet primære kvalitative data igennem interviews med relevante aktører, herunder med Frederik Bæk, IKT-koordinator på casestudiet (jf. afsnit 1.6), samt Michael Ørsted, afdelingschef for KPH-lufthavn og formand for bygherreforening.

De indsamlede data er først og fremmest blevet transskriberet og derefter analyseret ved brug af forskellige metoder, herunder problemtræ, mål-træ og Contextual design. Transskriberingerne er lagt til som bilag, så man i kraft heraf kan henvise til fremhævede citater, som forekommer igennem rapporten.

Herefter, er rå data fra interviews blevet analyseret ved brug af forskellige metoder henholdsvis til at kortlægge nuværende arbejdsprocesser i forbindelse med validering af bygningsmodeller og afklare dertilhørende problemstillingerne. Kvalitative data er således blevet anvendt til formål for at afklare problemets eksistens, omfang og validitet fra et empirisk perspektiv. I nedenstående belyses metoden for hvordan interviewet er blevet foretaget.

Interviewmetode

I forbindelse med indsamling af kvalitative data er der foretaget interview med udgangspunkt i bogen ”InterView” af Kvale & Brinkmann. (Kvale, Brinkmann and Forlag, 2009) Der er anvendt et narrativt semi-struktureret interviewform. Denne form danner et grundlag for at få beretninger fra respondenterne baseret på deres personlige og professionelle erfaringer. Interviewformen er således åben hvor retningen styres af svar som løbende forekommer i løbet af interviewet.



Tabel 2 Interviewmetode, Harboe (2001)

Derfor kan et narrativt interview være svært at styre. For at undgå at samtalen udvikler sig i irrelevante retninger, er interviewet blevet semi-struktureret igennem interviewguide (jf.

bilag). Interviewguiden fastlægger rammerne og indeholder spørgsmål, som ønskes besvaret. Interviewguiden er fremsendt til respondenterne forinden mødet.

1.8 Case: Lindbjergskolen (LBS)

Nærværende afsnit beskriver byggeprojektet som ligger til grund for rapportens undersøgelser, samt formålet med case-inddragelsen.

Herning Kommune investerer 200 millioner kroner i en ny folkeskole i Hammerum til ca. tusind børn. Skolen vil blive indrettet som en miniature by bestående af bygninger til indskoling, mellemtrin og udskoling. De tre bygninger bindes sammen af et åbent modtagelsesareal med indgang til skole, bibliotek og hal. (Licitation.dk, 2019)



Fakta om byggeriet:

Type: Folkeskole, kommunalt
Nuværende fase: Udførelse
Budget: 200 mio. kroner
Størrelse: 15.000 kvm
Byggeperiode: 2017 - 2020

Sagens aktører:

Arkitekt: Årstiderne Arkitekter
Landskabsarkitekt: Mutoxia APS
SWECO A/S
KPC Herning A/S

Byggesagen er blevet inddraget med henblik på at kortlægge og analysere måden hvorpå datagrundlaget i bygningsmodeller kvalitetssikres. Casestudiet er afgrænset til Årstiderne Arkitekter, hvor der fokuseres på hvordan de specificerer deres informationsbehov til øvrige parter samt hvordan sikrer, at modparten rent faktisk har moduleret i henhold til aftalen.

Casestudiet beskrives som unik single case, hvor der tilsigtes en ny kombination af viden og casen. Formålet med casestudiet er at analysere eksisterende proces med henblik på at kortlægge problemstillinger. Hertil udvælges en teori, som anskues at være afhjælpende i forhold til de opstillede problemstillinger. I løbet af rapportens kapitler vil den udvalgte teori analyseres, bearbejdes og konkluderes i forhold til casens problemstillinger.

1.9 Afgrænsning

Der indgår vidt forskellige mekanismer i byggeprojekter hvilket danner grundlaget for en begrundet afgrænsning. Derfor er undersøgelsesfeltet først og fremmest afgrænset i henhold til specialets faglige område: bygningsinformatik. Dernæst er der foretaget afgrænsning på baggrund af specialets emne: valideringen af bygningsmodeller. Yderligere er emnet afgrænset i kraft af en problemformulering. Det medfører, at byggeprojekter betragtes fra et IT-perspektiv. Alle mekanismer, der intet har at gøre med IT og rapportens problemformulering, er ud af fokusområdet.

Rapportens formål er at undersøge om IfcDoc-værktøjet kan effektivisere valideringsprocessen. IfcDoc har en indbygget valideringsfunktion, som sammenligner IFC-eksport i henhold til et defineret MVD. Målet er således at udvikle MVD baseret på LOD-DK-niveauer, som er specificeret i bygningsdelsspecifikationer. Den analyserende del er afgrænset i henhold til en enkel bygningsdel, da processen for MVD-udvikling er gældende for øvrige bygningsdele og deres egenskabsdata.

Tabel 3 LOD-DK-niveauer, som ligger til grund for udvikling af en MVD. (bygningsdelsspecifikationer)

BYGNINGSDEL LOD-DK	EGENSKABSDATA
RUM – 325	Rumnavn Rumnummer Areal Etagenavn
VÆG – 400	Typenavn Tykkelse Placering: Etage Konstruktionsopbygning
VINDUE - 325	Typenavn Bredde Højde Placering: Etage Flugtvej Opbygning

II. Analyserende

Kapitlet følger datamodelleringsprocessen, som belyst under metodeafsnittet. Analyserne i nærværende kapitel afklarer følgende underspørgsmål:

- Hvad er virkeligheden, som ligger til grund for konceptualisering?
- Hvordan abstraheres denne virkelighed i en konceptuel model?
- Hvordan realiseres det igennem MVD?

2.1 Virkelighed

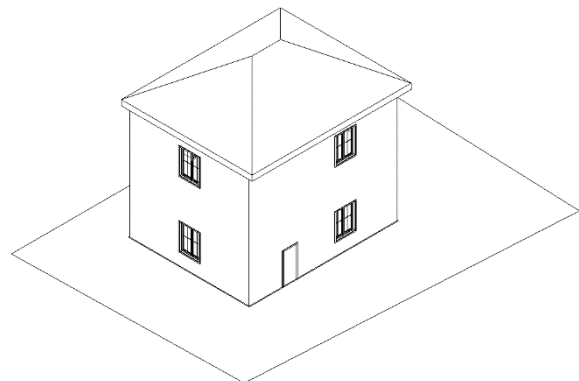
Udvikling af en MVD baseret på datamodelleringsprocessen, kræver først og fremmest en klar og konkret definition af virkeligheden. I rapportens kontekst, er virkeligheden en række egenskabsdata defineret igennem bygningsdelsspecifikationer (jf. afsnit afgrænsning).

Formålet med afsnittet er at danne en tilstrækkeligt grundlag for udvikling af en konceptuel model igennem afklaring af følgende underspørgsmål:

- Hvad er attributterne for enkelte egenskabsdata?
- Hvor i modellen befinder de sig?

Afklaring af ovenstående foretages på baggrund af en opstillet bygningsmodel. Modellens informationsindholdet fremgår af nedenstående tabel. Formålet med modellen er at fastlægge attributterne ved de enkelte egenskabsdata, derved danne grundlaget for en konceptuel model.

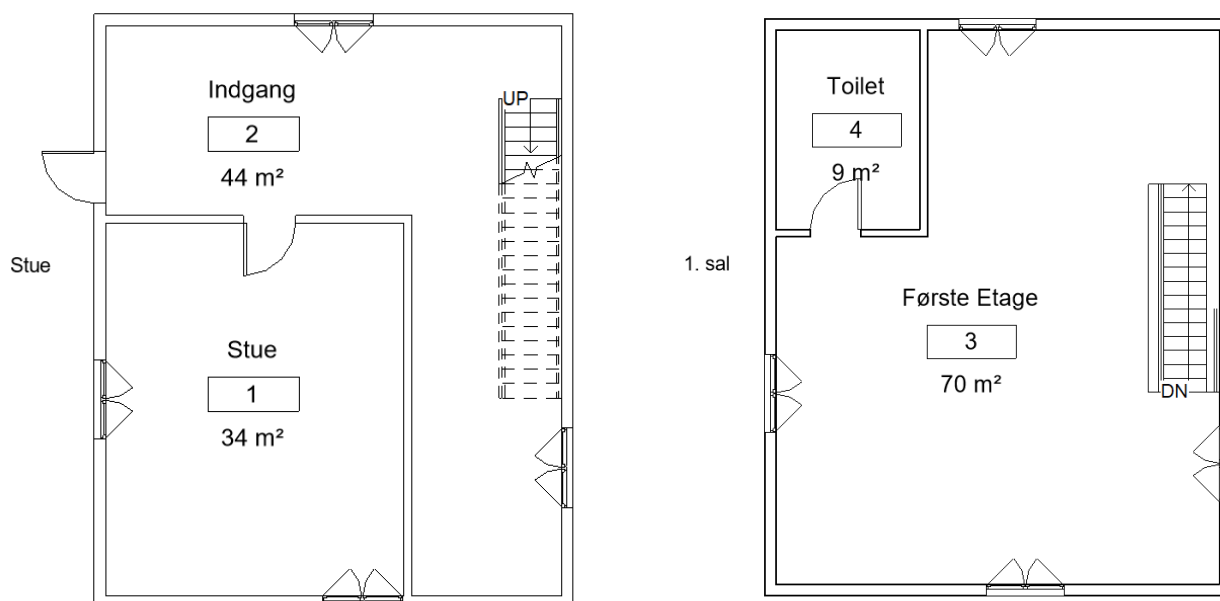
Bygningsdel	Level of detail (LOD DK)
Rum	325
Væg	400
Vindue	325



Figur 8 Opstillet "virkelighed", egenudviklet model i Revit

2.1.1 Rum LOD 325

Egenskabsdata for rum i bygningsmodellen



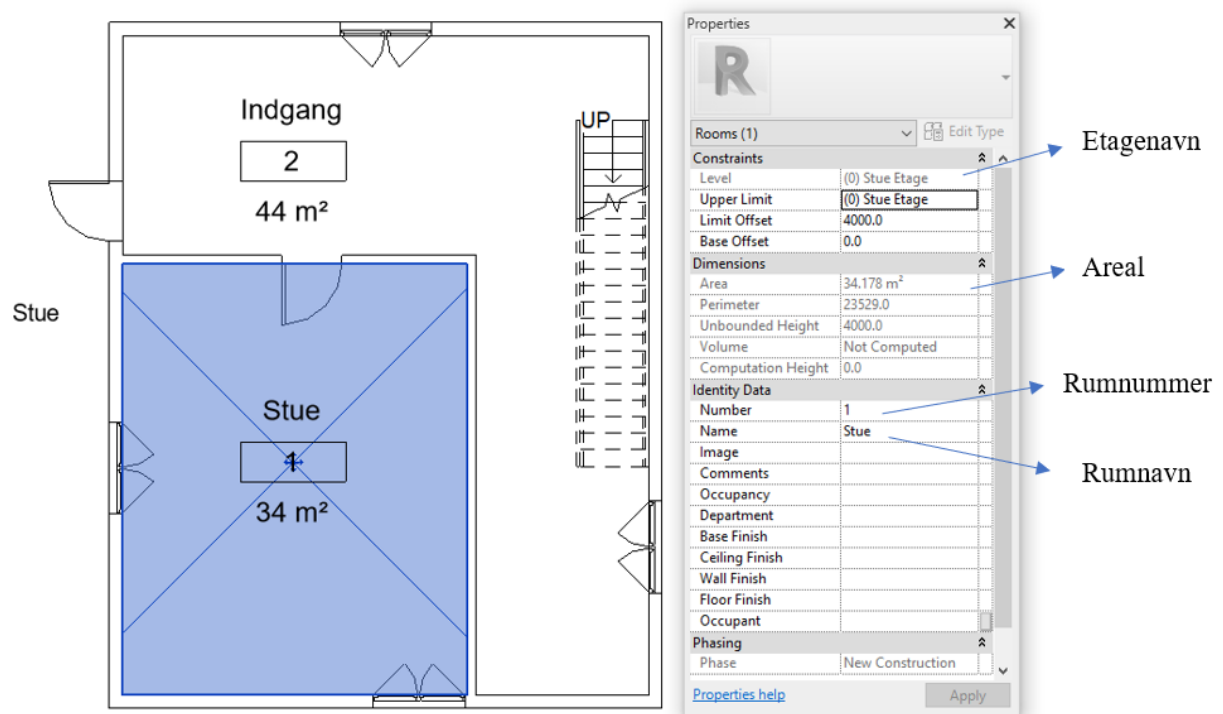
Figur 9 Rum LOD 325

Modellen består af i alt fire rum fordelt på to etager. Deres egenskabsdata beskrives som følgende:

Rumnavn	Rumnummer	Areal	Etagenavn
Stue	1	34	(0) Stue Etage
Indgang	2	44	(0) Stue Etage
Første Etage	3	70	(1) 1. sal
Toilet	4	9	(1) 1.sal

Tabel 4 Egenskabsdata for Rum

Lokalisering af egenskabsdata for rum



Figur 10 Type A: Attributterne for egenskabsdata findes under vinduet: Properties

Figuren viser, at attributterne for de enkelte egenskabsdata tilknyttet Rum – LOD 325 befinder sig i vinduet Properties. For eksempel kan det ses, at parameteret **Rumnavn** befinder sig under vinduet Properties, egenskabsstypen: Identity data under parameteren Name.

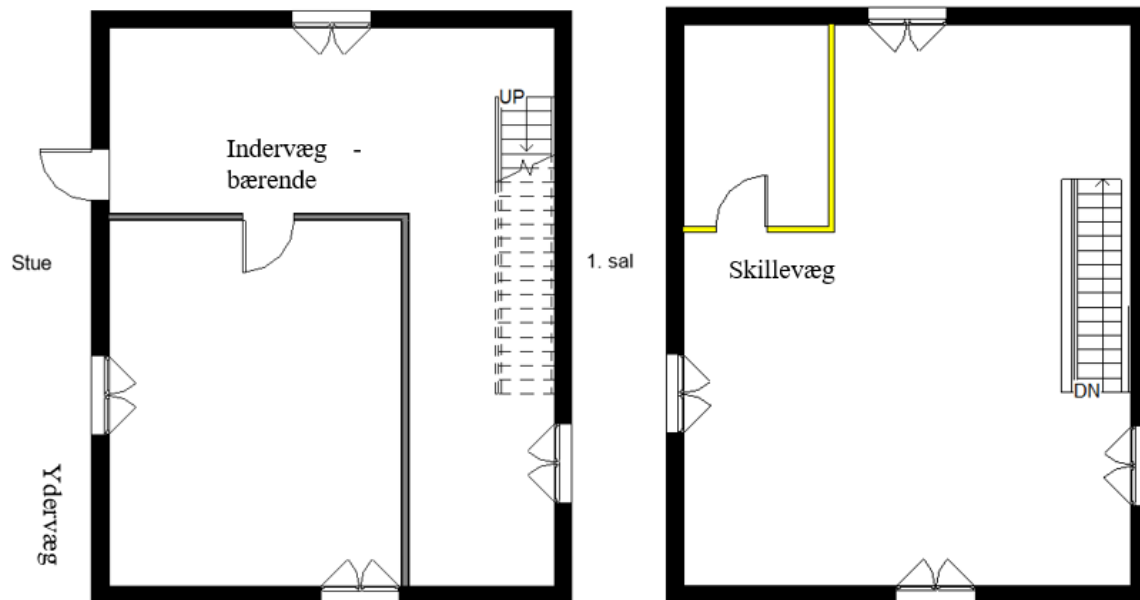
Egenskabsdata:	Rumnavn	Rumnummer	Areal	Etagenavn
Egenskabssæt	Identity Data	Identity Data	Dimensions	Constraints
Egenskabsnavn	Name	Number	Area	Level

Tabel 5 Lokalisering af egenskabsdata for rum 325

Det vil sige, at Rum LOD 325 DK, medfører at der er indtastet værdi, som vist på figuren for samtlige Rum i en bygningsmodel.

2.1.2 Væg LOD 400

Egenskabsdata for rum i bygningsmodellen



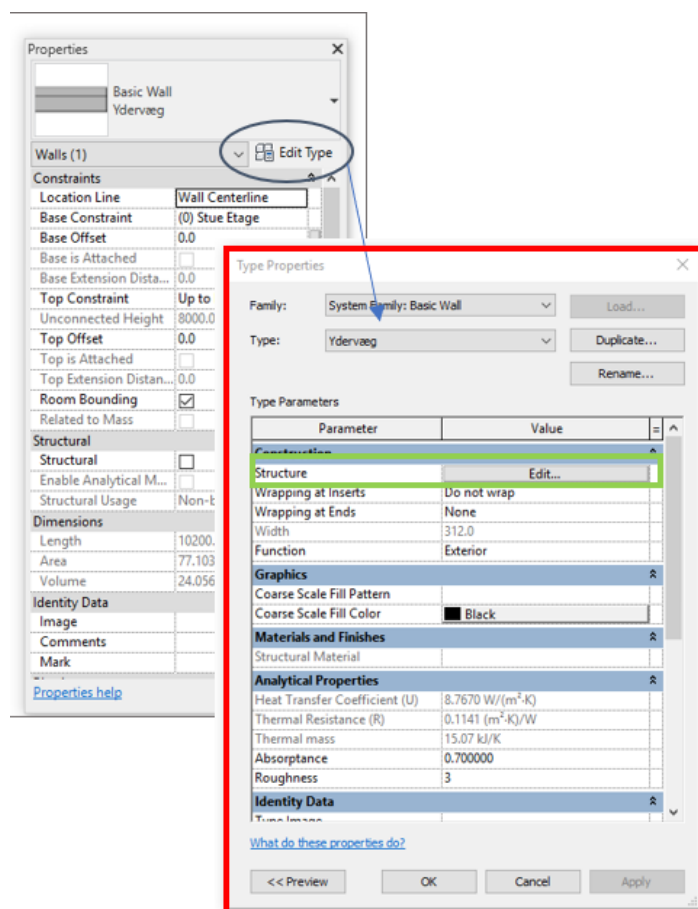
Figur 11 Vægge i modellen

I bygningsmodellen er der tre forskellige vægtyper. Disse forekommer i farven sort, grå og gult i ovenstående figur. Egenskabsdata for væggene konkretiseres i forhold til deres værdi i nedenstående tabel-

Typenavn	Tykkelse	Placering: Etage	Konstruktionsopbygning
Ydervæg	312	Stue	Beton: 100 mm Isolering: 200 mm Gipsplade: 12 mm
Indervæg - Bærende	131	Stue	Gipsplade: 12 mm Beton: 100 Gipsplade: 12 mm
Skillevæg	95	1.sal	Gipsplade: 12 mm Beton: 64 Gipsplade: 12 mm

Figur 12 Egenskabsdata over vægge

Lokalisering af egenskabsdata for væg



Figur 13 Type B: attributterne for egenskabsdata findes under Type Properties, Type C: er markeret med grønt

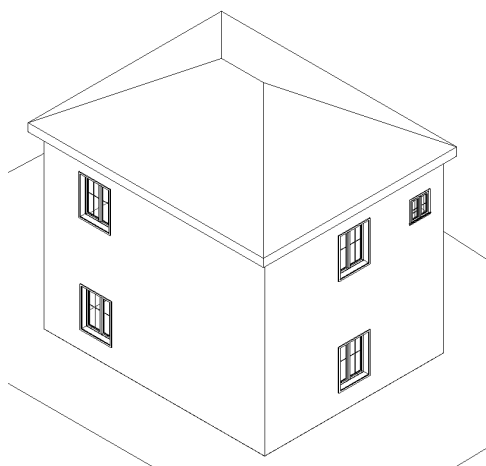
Figuren illustrerer, at egenskabsdata Typenavn, befinder sig under Properties > Type Properties, under parameternavnet type. Vinduet som er markeret med rødt, tildeles navnet egenskabssæt, vinduet som er markeret med grønt, kalder vi egenskabsparameter. Denne opdeling gør det nemmere at lokalisere øvrige egenskabsdata som vist i nedenstående tabel.

Egenskabsdata	Typenavn	Tykkelse	Placering: Etage	Konstruktionsopbygning
Egenskabssæt	Type Properties	Construction	Constraints	Construction
Egenskabsnavn	Type	Width	Base Constraints	Structure

Tabel 6 Lokalisering af egenskabsdata for vægge LOD 400

2.1.3 Vindue LOD 325

Egenskabsdata for vindue

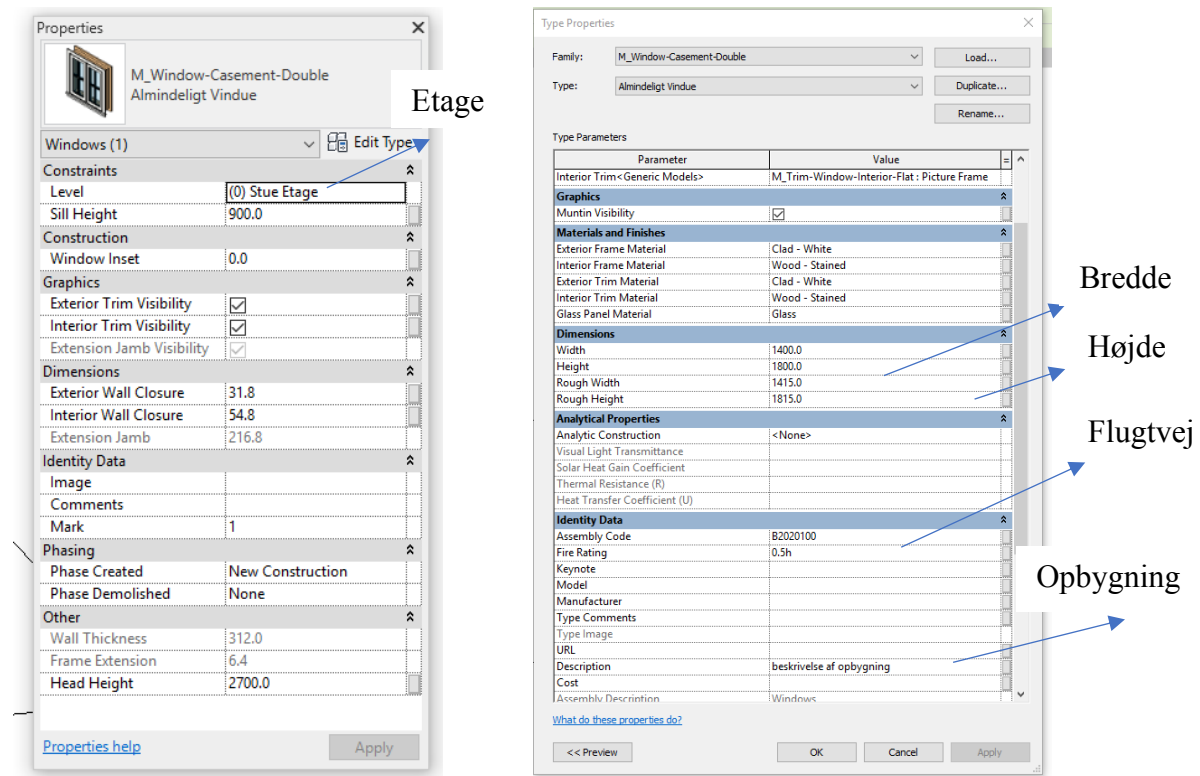


Figur 14 Vinduer i bygningsmodellen

Modellen består af i alt 9 vinduer, fordelt over to typer. Egenskabsdata for typerne fremgår af nedenstående tabel.

Typenavn	Bredde: Hulmål	Højde: Hulmål	Placering: Etage	Flugtvej	Opbygning
Badeværelse vindue	915	865	Første sal	0.5 h	”beskrivelse”
Almindeligt vindue	1415	1815	Stue og Første sal	0.5 h	”beskrivelse”

Lokalisering af egenskabsdata for vindue



Figur 15 Egenskabsdata for vindue

Egenskabsdata	Bredde: Hulmål	Højde: Hulmål	Placering: Etage	Flugtvej	Opbygning
Egenskabssæt	Dimensions	Dimensions	Constraint	Identity Data	Identity Data
Egenskabsparameter	Rough Width	Rough Height	Level	Fire Rating	Description

Tabel 7 Lokalisering af Egenskabsdata over vindue

Delkonklusion

Attributterne for egenskabsdata inddeles i tre typer afhængig af deres placering.

- Type A – Attributterne findes under Properties
- Type B – Attributterne findes under Type Properties
- Type C – Attributterne findes under et tredje vindue

Nedenstående tabeller danner oversigt over attributterne ved de enkelte egenskabsdata tilknyttet bygningsdelene.

Rum	Type A			
Egenskabsdata	Rumnavn	Rumnummer	Areal	Etagenavn
Egenskabstype	Identity Data	Identity Data	Dimensions	Constraints
Egenskabsparameter	Name	Number	Area	Level

Væg	Type A	Type B		Type C
Egenskabsdata	Placering	Typenavn	Tykkelse	Konstruktionsopbygning
Egenskabstype	Constraints	Type Properties	Construction	Construction
Egenskabsparameter	Base Constraints	Type	Width	Structure

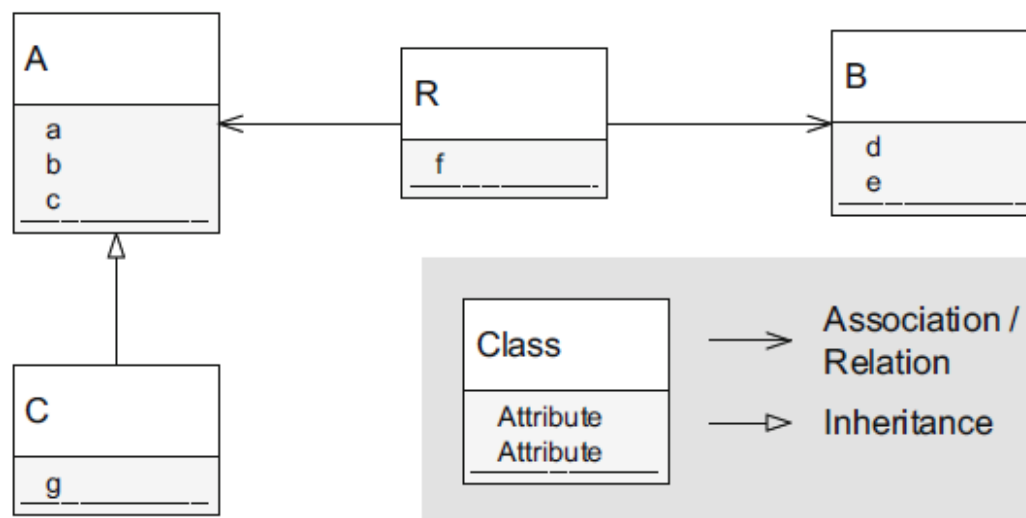
Vindue	Type A	Type B			
Egenskabsdata	Placering: Etage	Højde	Bredde	Flugtvej	Opbygning
Egenskabstype	Constraints	Type Properties	Construction	Identity Data	Identity Data
Egenskabsparameter	Base Constraints	Type	Width	Fire Rating	Description

Tabellerne skal forstås sådan, at for eksempel egenskabsdata Højde for vinduet forekommer i en Type B – vindue (Type Properties, jf. fig. 15) under egenskabstypen Constraint og parameteren Base Constraints. Inddelingen gør det nemmere at udvikle en konceptuel model og en Model View Definition. (uddybes under Del III)

2.2 Konceptualisering: Unified Modelling Language (UML)

*"A conceptual Model is a Representation of a System, made of the composition of concepts, which are used to help people know, understand, or simulate a subject the model represents."*⁷

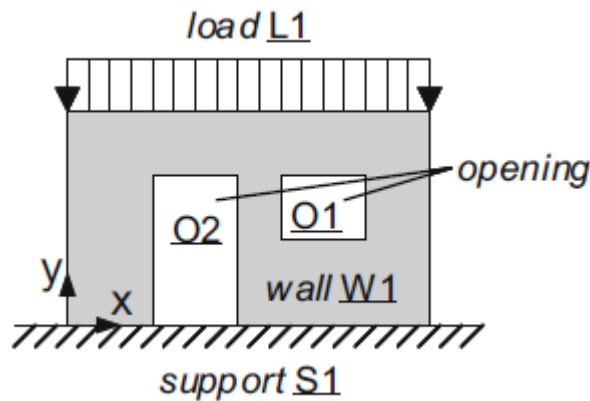
Unified Modelling Language (UML) bruges til konceptualisering af objektorienteret datamodeller. UML-konceptualisering danner grundlaget for udvikling af en datamodel. Denne abstraherer en virkelighed og beskriver koncepterne ud fra grafiske notationer. Nærværende afsnit analyserer på UML med henblik på at beskrive LOD-niveauerne igennem en konceptuel model, derved skabe et bedre grundlag for udvikling af MVD. Det teoretiske grundlag for UML, er udarbejdet med udgangspunkt i Koch & König, 2018.



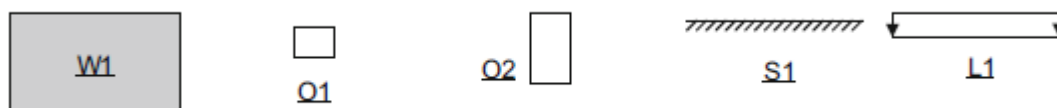
Figur 16 Class-diagram (koch & konig, 2018)

UML er en international standard notation, som bruges til at beskrive objektorienteret datamodeller (OOM) igennem tekst såvel som grafiske symboler og diagrammer. De vigtigste diagramtyper er en såkaldt "Class Diagram". Disse fremgår af figuren som A, B, C og R, og beskriver koncepter, som indgår i virkeligheden. Enkelte Class diagrammer har attributterne a, b, c, d, e, f, g, samt en Relation til øvrige Classes, som illustreres med pile.

⁷ https://en.wikipedia.org/wiki/Conceptual_model



Figur 17 Eksempel væg



Figur 18 Entities (objekter) som indgår i væggen

Figuren viser en væg, der har to åbninger, som hviler på en grund, hvor denne er udsat for last. Ovenstående figur er et simplificeret eksempel, som er inddraget med henblik på at illustrere hvordan denne realiteten konceptualiseres igennem UML-notation.

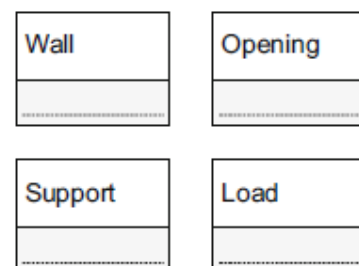
2.2.1 Entities og Entity Types

En "Entity" er også kendt som en "Object" eller "Instance". Denne er en specifik data element som repræsenterer et objekt fra virkeligheden. Entities kan repræsentere både materialer med fysiske form, såsom væg, søjle, vindue og ikke-fysisk form såsom rum, arealer eller last. Hver Entity er defineret igennem dens identitet, tilstand samt form.

I figuren med væggen, kan virkeligheden abstraheres igennem følgende Entities:

1. Væggen W1
2. Åbninger O1 og O2
3. Underlag S1
4. Lasten L1

UML Class Diagram



Figur 19 UML-Notation for Entity Types (Class)

En "Entity Type" også kaldet "Class", klassificerer Entities, som deler samme struktur og karakter. Denne kan være form, formål eller udseende. Entity Type danner en slags template, som danner basis for udvikling af enkelte Entities. På denne måde, kan det siges, at en Entity er oftest en "Instance" af en Entity Type.

I eksemplet med væggen, har vi brug for fire Entity Type for at modellere væggen: Væg, åbninger, underlag og last. W1 er således en Instance af væg, som altså er en Entity type. Entity types moduleres igennem UML-Class Diagram, som vist på figur 12.

2.2.2 Attributter

Et Entity identificeres igennem dens Attributter. Attributterne angiver Entity dens egenskaber, samt relaterede informationer og data. Entities som har de samme entity typer deler de samme attributter, men varierer i forhold til værdien i attributterne. Attributterne defineres derfor i entity type, og har et navn og en datatype. Datatype specificerer typen af information, som kan tilknyttes attributten. Overordnet skelnes mellem en primitiv (atomic) types, Composite types og Data Structures.

Primitive datatyper består oftest af

- Integer eller heltal
- Real numbers
- Boolean eller logiske værdier
- Charecters

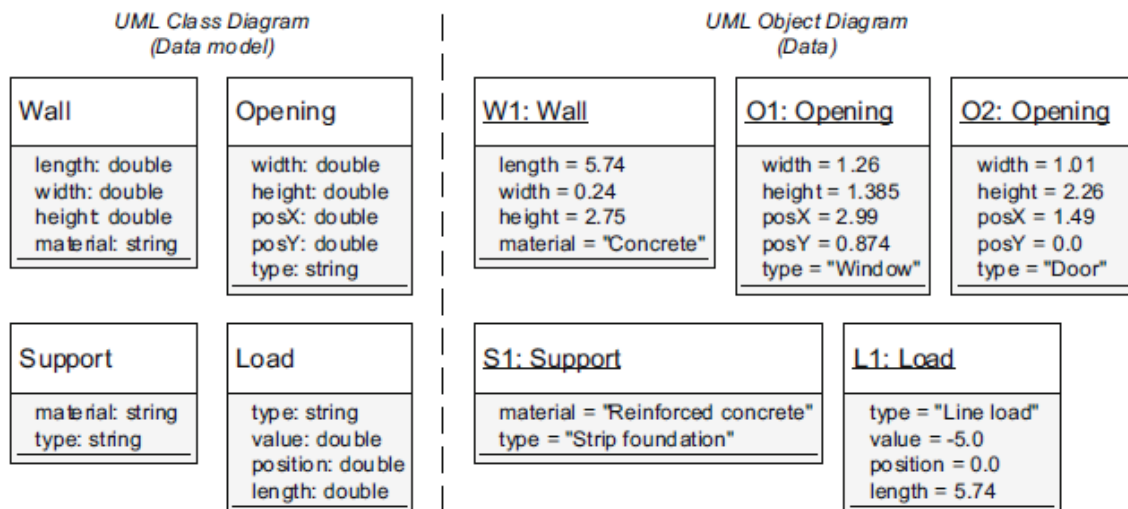
Composite types er en sammensætning af Primitive datatyper. For eksempel, en text som består af en sekvens af karakter, eller en Array af værdier af samme primitive typer. I modsætning til Array, bruges en Enumeration til at definere distinktive værdier af samme primitive data typer. Den faktiske værdi af en Enumeration er en af de definerede værdier. (se nedenstående tabel)

Type category	Data type	Example definitions (DEF) and values (VAL)
Primitive types	Integer number (INT, INTEGER, LONG)	VAL: -123, 0, 2, 875
	Real number (FLOAT, DOUBLE)	VAL: -1.234, 1.234e02
	Boolean/logical value (BOOL, BOOLEAN, LOGICAL)	VAL: true (0), false (1)
	Character (CHAR, CHARACTER)	VAL: "a", "1", "α", "@", "≤"
Composite types	Text (STRING)	VAL: "abc", "123", "a1@_x"
	Enumeration (ENUM, ENUMERATION)	DEF: Color := {blue, green, red, yellow}; VAL: Color.green
	Array, series or sequence (ARRAY), finite index-based series of values of the same primitive data type	DEF: 3D_Vector := ARRAY(1..3) of DOUBLE; VAL: [-1.23, 4.56e-5, 123.45]
Data structures	Entity or Class (ENTITY, CLASS), finite number of attributes of different data types (primitive or composite type, data structure)	DEF: ENTITY/CLASS Date := {day:INT, month:INT, year:INT}; VAL: {15, 2, 2012}
	List or sequence (LIST), (in-)finite index-based series of values of the same data type or structure	DEF: List_of_Openings := LIST of CLASS(Opening); VAL: [O1, O2, O3]
	Set (SET), (in-)finite unsorted set of values of the same data type or structure	DEF: Set_of_Openings := SET of CLASS(Opening), VAL: {O2, O1, O3}

Figur 20 forskellige datatyper

En **data structure** opfanger forskellige data typer som en enhed. Datatyperne som anvendes, kan være både Primitive, Composite eller andet datastruktur. Ovenstående tabel danner oversigt over de forskellige data typer.

Hvis vi vender tilbage til vores væg eksempel, kan vi definere forskellige attributter for vores Entity Types (classes) og tildele specifikke værdier til attributterne for enkelte Entities (objekter).



Figur 21 attributter i object-orienterede datamodeller: datamodel (venstre) og data (højre)

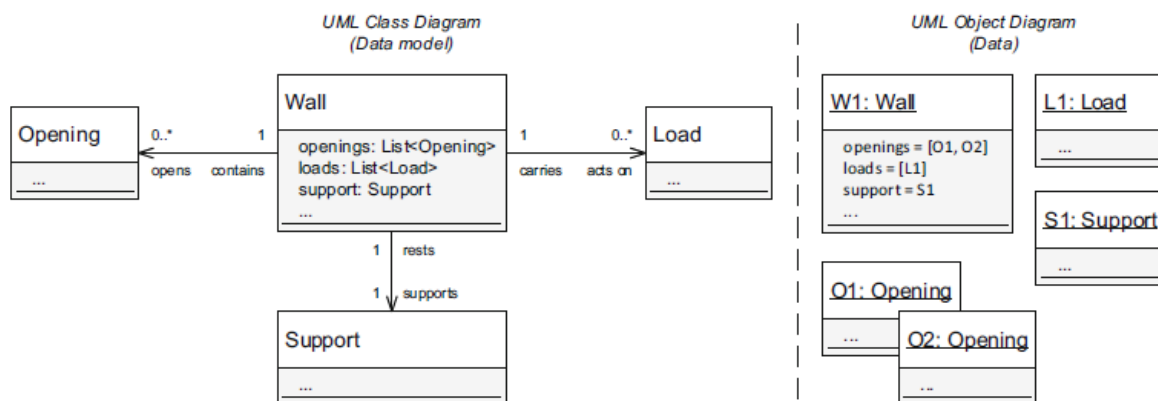
Ovenstående figur illustrer datamodel for væggen til venstre, samt hvordan data realiseres baseret på datamodellen til højre. Som vist til venstre i figuren, har Entity Typerne: væg, åbning, underlag og last fået tildelt attributter. For eksempel, definerer Entity Typen ”åbning” følgende attributter: bredde, højde, posX og posY. Disse attributter bruger henholdsvis datatyperne Double og String. Den højre side i figuren viser UML Object Diagram, som indeholder data, der indgår i eksemplet med væggen: fem objekter, W1, O1, O2, S1 og L1, samt attributterne til de enkelte objekter. For eksempel, åbningen O1 er karakteriseret igennem værdierne til attributter, hvor bredde = 1.26, højde 1.385, posX = 2.99, posY = 0874 og type = ”window”.

2.2.3 Relationer (Associations)

Relations eller Associations er datamodelleringskoncepter, som beskriver forholdet eller afhængigheden mellem Entities (objekter). Behovet for modellering af relationer opstår når en Entity indeholder informationer, som en anden Entity har brug for. Denne kaldes for ”data dependency”.

Relationer kan enten være Binary eller Cardinalities. Binary Relations er den mest udbredte relationstype. Denne beskriver et forhold mellem to Entities (objekter). Cardinalities derimod, beskriver hvor mange antal Entities på den ene side har et forhold til hvor mange antal entities på den anden side. Yderligere skelnes der mellem Directed (direkte) og (indirekte) relationer.

Objekt-orienteret modellering igennem UML, er relationer mellem objekter defineret igennem brug af Attributer. Nedenstående figur viser et eksempel herpå-



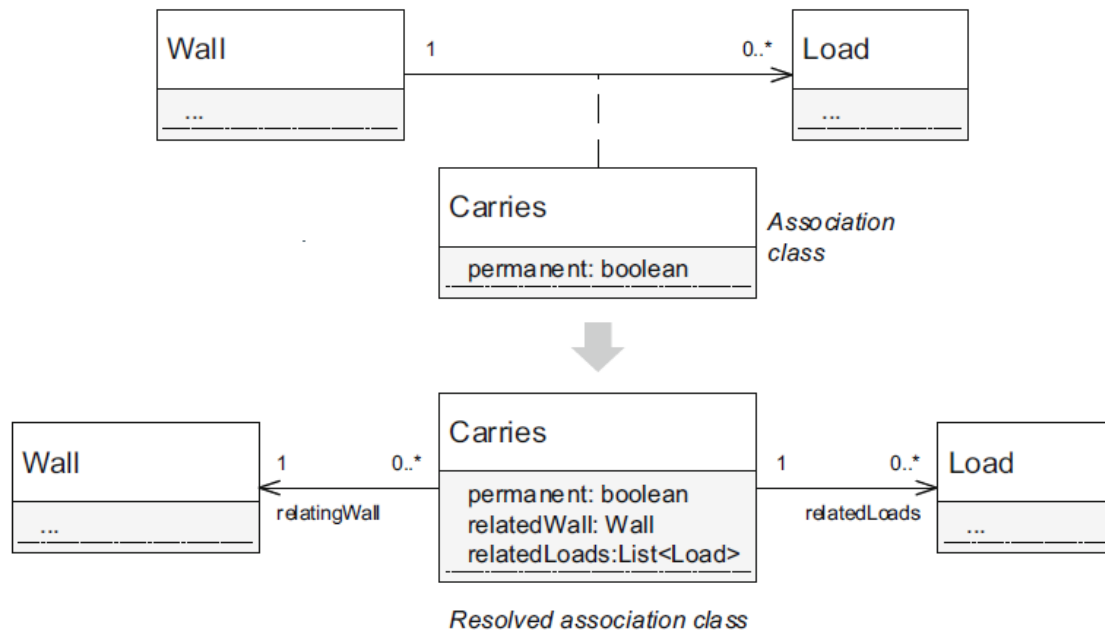
Figur 22 Relationer i Objectoriented modeling – datamodel til venstre, data til højre

Relationer kan blive refereret enten direkte eller igennem en såkaldt Association Class. Forskellen på disse former for tilknytning af relationer er, at man igennem anvendelse af Association Class, kan tilføje yderligere egenskaber til en attribut.

Hvis vi tager udgangspunkt i eksemplet med væggen igen, så er objektet Væg associeret med øvrige objekter, herunder åbning og last. Denne illustreres til højre i ovenstående figur. UML angiver relationer med solide linjer og en åben pil mellem objekterne. Kardinaliteterne er angivet i hver sin ende af pilene. Til højre for figuren er der for eksempel angivet et 1-tal for hver sin ende i pilen mellem objektet Wall og et 0-tal for Opening. Det betyder, at netop én væg er associeret med et vilkårligt antal (0..*) af Opening. Denne form for relation er defineret i classdiagrammet for Wall som attributen openings, der består af datatype List<opening> som lagre relaterede åbninger i en liste. På samme måde, defineres relationen mellem Wall og Load. Wall-Support relationen er moduleret igennem attributten support for type Support.

UML-diagrammet til højre præsenterer realisering af de relationer mellem objekter som indgår i eksemplet. I modsætning til figur 14, beskriver figur 15 yderligere attributter og deres værdier med hensyn til enkelte Entities. For eksempel, Wall er associeret med to åbninger, O1 og O2 implementeret igennem attributter værdierne O1, O2.

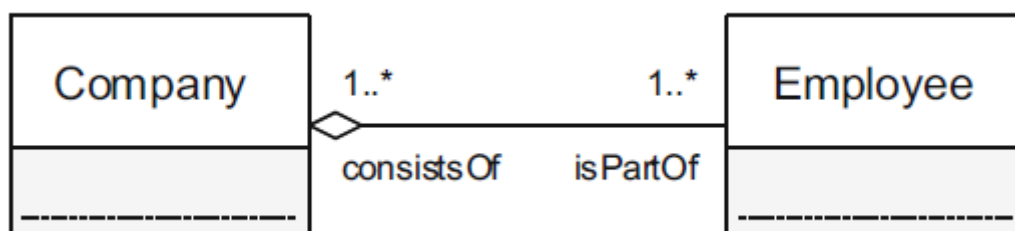
Øverst i figur 16, beskrives en dedikeret Association class "Carries". Denne bruges til at modellere relationer mellem Væg (Wall) og last (Load) objekter, og tilføjer yderligere information (semantic) til forholdet mellem disse to objekter. Denne information er moduleret som attributten "permanent" ved brug af datatypen Boolean. Denne afgør om lasten er permanent (true) eller midlertidig (false). Så længe formålet er at implementere denne del af datamodellen i en object-orienteret software, skal samtlige association class løses. Det betyder, at den intenderende relation mellem Wall og Last forekommer i en ny class Carries, hvor de to forhold, Wall-Carries og Carries-Load, inkluderer korresponderende attributter. Denne er afbildet i den nederste del i figuren.



Figur 23 Relation class i object-oriented model og dens resolution.

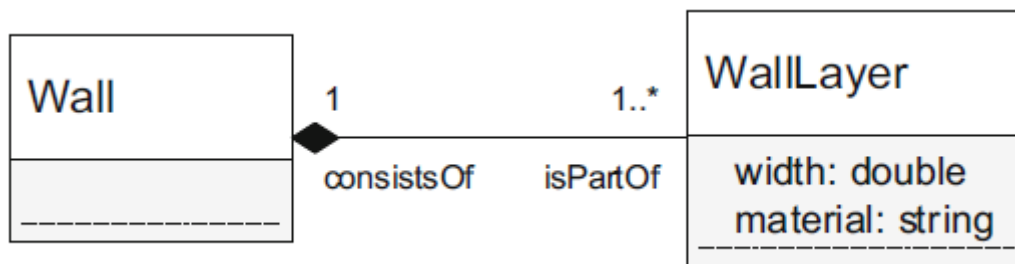
Aggregations og Compositions (relationstyper)

Aggregation (aggregering) og Composition (sammensætning) er særlige typer af relationer og beskriver Whole-Part (hel-del) relationer mellem objekter. Disse typer af forhold beskrives igennem relationer "is-part-of" (en-del-af) eller "consists-of" (består-af). En Entity eller objekt repræsenterer hel (aggregate) og en aggregeret entity eller objekt repræsenterer del af helheden. I dette sammenhæng definerer "Composition" en slags stærk sammensætning, hvor delene kan ikke eksistere uden helheden. I objekt-orienterede modellering, er der dedikerede symboler og koncepter i UML-class diagrammer til at udtrykke relationstyperne Aggregation og Composition.



Figur 24 Aggregation

Figur 17 er et eksempel på en Aggregation relation. Figuren illustrerer, at ansatte kan eksistere uafhængig af virksomheden. Man bruger Aggregation, til at beskrive relationer hvor delen ikke er afhængig af helheden. Figuren viser, at en virksomhed består af mindst en (1) eller mange (*) ansatte. Og at medarbejderne er en del af mindst en (1) eller flere (*) virksomheder. I UML-class diagrammer angives Aggregation-relationer med en tom diamant-formet symbol ved enden af helheden.



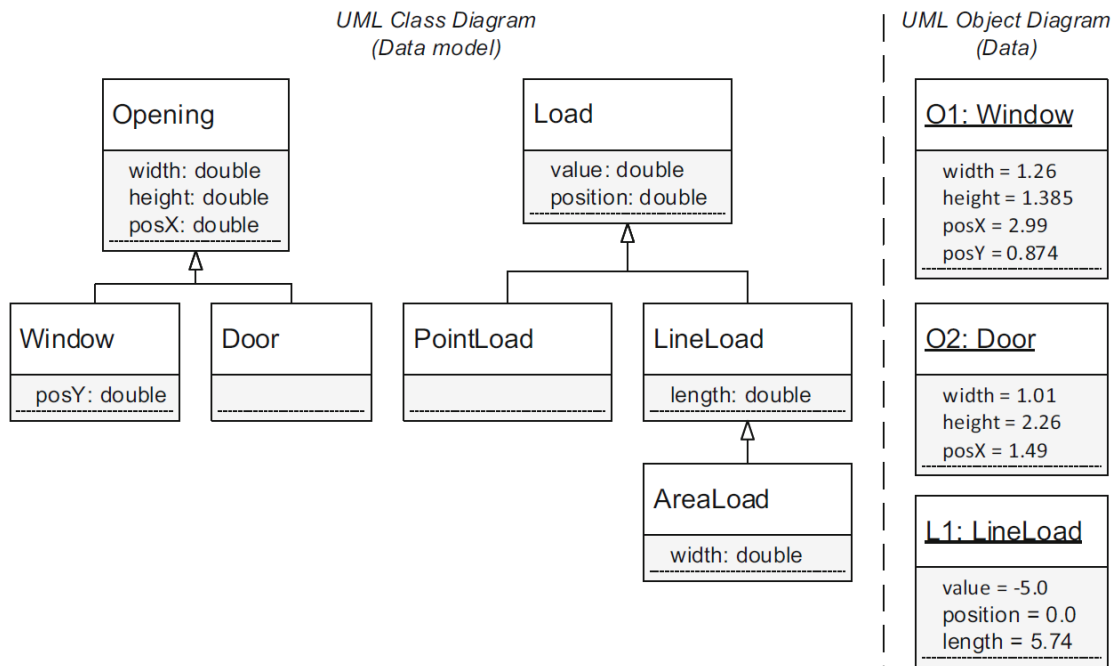
Figur 25 Composition

I den tidligere eksempel med væggen, kan det antages at væggen består af flere dog mindst en lager (1..*). Denne relation kan moduleres som en Composition, fordi enkelte lag i væggen, eksempelvis isoleringslaget, kan ikke eksistere uden væggen. Figur 18 illustrere en Composition-relation. I UML angives denne slags relation med en fyldt diamant-formet symbol for enden af helheden. WallLayer definerer attributter som repræsenterer hvert enkelt lager i forhold til tykkelse eller materiale.

Inheritance (relationstype)

Relationer mellem Classes (klasser) spiller en vigtig rolle i forhold til beskrivelsen af semantiske informationer i datamodeller. I denne kontekst er der en anden essentiel datamodelleringskoncept ved navn "Inheritance", også kendt som "Specialization and generalization". Inheritance tillader definering af specialiserede Classes (sub-classes, child classes), samt generaliserede classes (super-classes, parent classes). Ideen bygger på, at Sub-classes (sub-klasser) kan inherit (arve) attributter eller egenskaber fra relaterede super-klasser og samtidig definere ekstra attributter som er en specialisering af super-class. Denne fungerer ligeledes den anden vej rundt, ved at super-classes repræsenterer generaliseringer af tilhørende sub-classes. Denne koncept skaber en hierarkisk klassifikations system (taksonomi) inden i datamodellen. I objektorienteret datamodellering kan systemet bruges til at udvikle et hierarkisk klassifikations struktur.

I objekt-oriented modeling kan inheritance klassificeres som en Single Inheritance eller Multiple Inheritance. Single inheritance betyder at hver enkelte sub-klass er associeret med præcis en enkelt super-class. Ved dette tilfælde bliver grafen til en slags familietræ. Multiple Inheritance derimod, tillader at sub-klasser kan have flere associerede super-klasser. Dette tilfælde er oftest problematisk, da konflikter kan opstå når sub-klasser arver attributter af samme navn fra flere super-klasser.



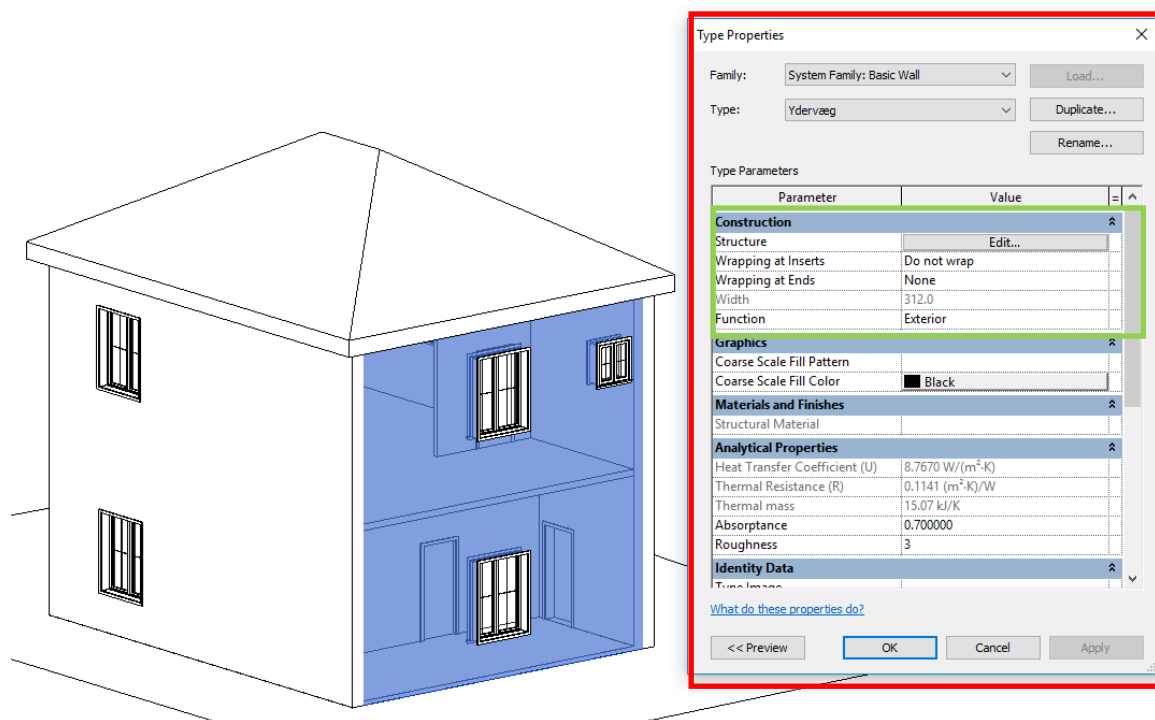
Figur 26 Inheritance i Object-Oriented modellering datamodel (venstre) & data (højre)

For at vende tilbage til eksemplet med væggen, kan objekter O1 og O2 have samme egenskaber, men differentier i forhold til type af åbningen. Indtil videre, er denne difference moduleret igennem attribute type i Class Opening, som vist på figur 19. Men med denne løsning, er det kun objektet som ved om den repræsenterer et vindue eller en dør. Datamodellen afslører semantisk information herom. Derfor er det mere passende at klassificere åbninger igennem Inheritance-konceptet. I sammenhæng med væggen, bør åbninger beskrives igennem en generalisering, hvor vinduer og døre er specialisering af super-klassen opening. Denne erstatter attribute typen. Så længe døren altid er placeret nederst i væggen (posY=0), behøver attributten kun at blive specificeret i class for Window og ignoreret for Door. Denne er illustreret til venstre i figur 19. Det er værd at bemærke at en Inheritance relation er symboliseret med en åben pil, som peger mod super-classen. UML-objektdiagrammet til højre illustrerer at O1 og O2 er henholdsvis Window og Door, som er sub-class til Opening.

Delkonklusion: Konceptualisering i UML

UML-diagrammer anvendes til konceptualisering af objekt-orienterede datamodeller. UML abstraherer en virkelighed i Class diagrammer, hvor alle objekter der indgår i virkeligheden repræsenteres igennem Classes, herunder sub-class og super-class. Alle Class har attributter, og attributterne definerer også Relationer til øvrige Classes.

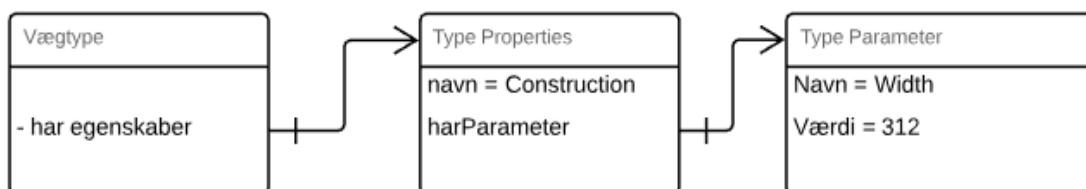
Normalt bruges UML til at beskrive hvordan en objekt-orienteret datamodel skal designs. I dette tilfælde eksisterer data. De fremgår af den definerede bygningsmodel i Revit. Igennem UML-konceptualisering af eksisterende data, kan datamodellen rekonstrueres.



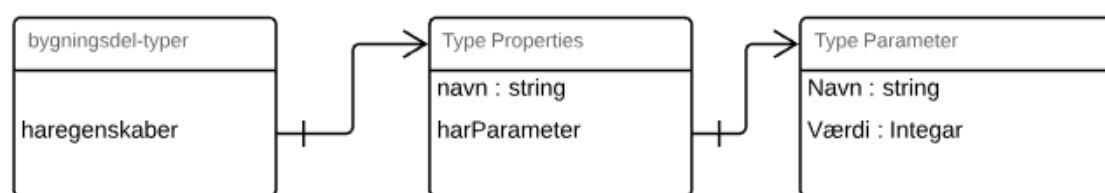
Figur 27 Rekonstruering af en datamodel ud fra eksisterende data.

Figuren viser, at Væg har vinduet Type Properties, som har Egenskabsstypen Construction. Denne virkelighed abstraheres, som vist i nedenstående UML-diagram.

Figur 28 UML-diagram over egenskabsdata Tykkelse for vægge



Ovenstående diagram beskriver Attributten Width igennem Classes, Relationer og attributter. På baggrund af data-diagrammet, kan det vurderes, at øvrige attributter som forekommer i Type B, kan beskrives ud fra generalisering-konceptet. Nedenstående datamodel er således gældende for alle egenskabsdata hvor attributterne forekommer i samme vindue.

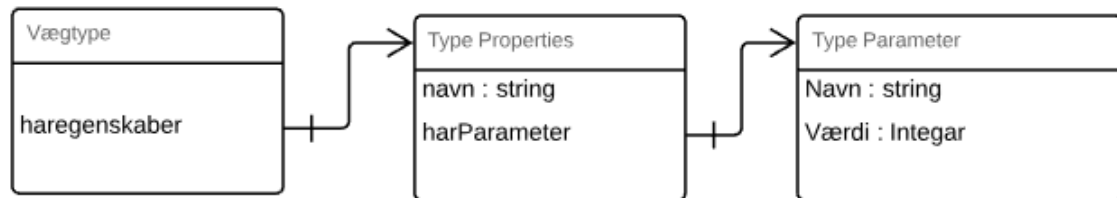


Figur 29 Datamodel for Type B egenskabsdata (jf. afsnit 2.1)

På baggrund af ovenstående Class-diagram, kan en MVD ved at finde frem til hvad Bygningsdeltyper, type Properties og Type Parameter hedder.

2.3 Realisering: Udvikling af MVD baseret på konceptuel model

Industry Foundation Class (IFC) er en datamodel defineret igennem EXPRESS. Model View Definition udgør en delmængde i Ifc-skemaet. (buildingsmart, 2018) Nærværende afsnit, analyserer på XML-Eksport af en bygningsdel for at oversætte nedenstående datamodel til EXPRESS.



Figur 30 UML-diagram over vægtype

- Vægtype
- Type Properties
- Type Parameter

Denne løses ved eksport af en væg, som vist til højre. Der tilføjes en værdi til parameteret Type Mark. Formålet er at lokalisere denne Parameter i XML-fil, for derved at finde frem til hvad de enkelte Classes som fremgår af datamodellen hedder.

The screenshot shows the 'Type Properties' dialog box. It has a 'Family' dropdown set to 'System Family: Basic Wall' and a 'Type' dropdown set to 'Generic - 200mm'. Below these are buttons for 'Load...', 'Duplicate...', and 'Rename...'. The 'Type Parameters' section contains a table with columns 'Parameter' and 'Value'. The 'Identity Data' section is expanded, showing various parameters like 'Type Image', 'Keynote', 'Model', 'Manufacturer', 'Type Comments', 'URL', 'Description', 'Assembly Description', 'Assembly Code', 'Type Mark' (with a red underline and the value 'textbox'), 'Fire Rating', and 'Cost'. At the bottom, there is a link 'What do these properties do?' and buttons for '<< Preview', 'OK', 'Cancel', and 'Apply'.

Parameter	Value
Heat Transfer Coefficient (U)	
Thermal Resistance (R)	
Thermal mass	
Absorptance	0.700000
Roughness	3
Identity Data	
Type Image	
Keynote	
Model	
Manufacturer	
Type Comments	
URL	
Description	
Assembly Description	
Assembly Code	
Type Mark	textbox
Fire Rating	
Cost	

```

<IfcWallType id="i1854">
  <GlobalId>3kSL0VGKv3gxJCujeqtuWE</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" ref="i1677"/>
  </OwnerHistory>
  <Name>Basic Wall:Generic - 200mm</Name>
  <IfcPropertySet id="i1926">
    <GlobalId>3kSL0VGKv3gxJCvDiqtuWE</GlobalId>
    <OwnerHistory>
      <IfcOwnerHistory xsi:nil="true" ref="i1677"/>
    </OwnerHistory>
    <Name>Identity Data</Name>
    <HasProperties exp:cType="set">
      <IfcPropertySingleValue id="i1886">
        <Name>Assembly Code</Name>
        <NominalValue>
          <IfcText-wrapper></IfcText-wrapper>
        </NominalValue>
      </IfcPropertySingleValue>
      <IfcPropertySingleValue id="i1887">
        <Name>Assembly Description</Name>
        <NominalValue>
          <IfcText-wrapper></IfcText-wrapper>
        </NominalValue>
      </IfcPropertySingleValue>
      <IfcPropertySingleValue id="i1888">
        <Name>Type Mark</Name>
        <NominalValue>
          <IfcText-wrapper>testxx</IfcText-wrapper>
        </NominalValue>
      </IfcPropertySingleValue>
      <IfcPropertySingleValue id="i1889">
        <Name>Type Name</Name>
        <NominalValue>
          <IfcText-wrapper>Generic - 200mm</IfcText-wrapper>
        </NominalValue>
      </IfcPropertySingleValue>
    </HasProperties>
  </IfcPropertySet>

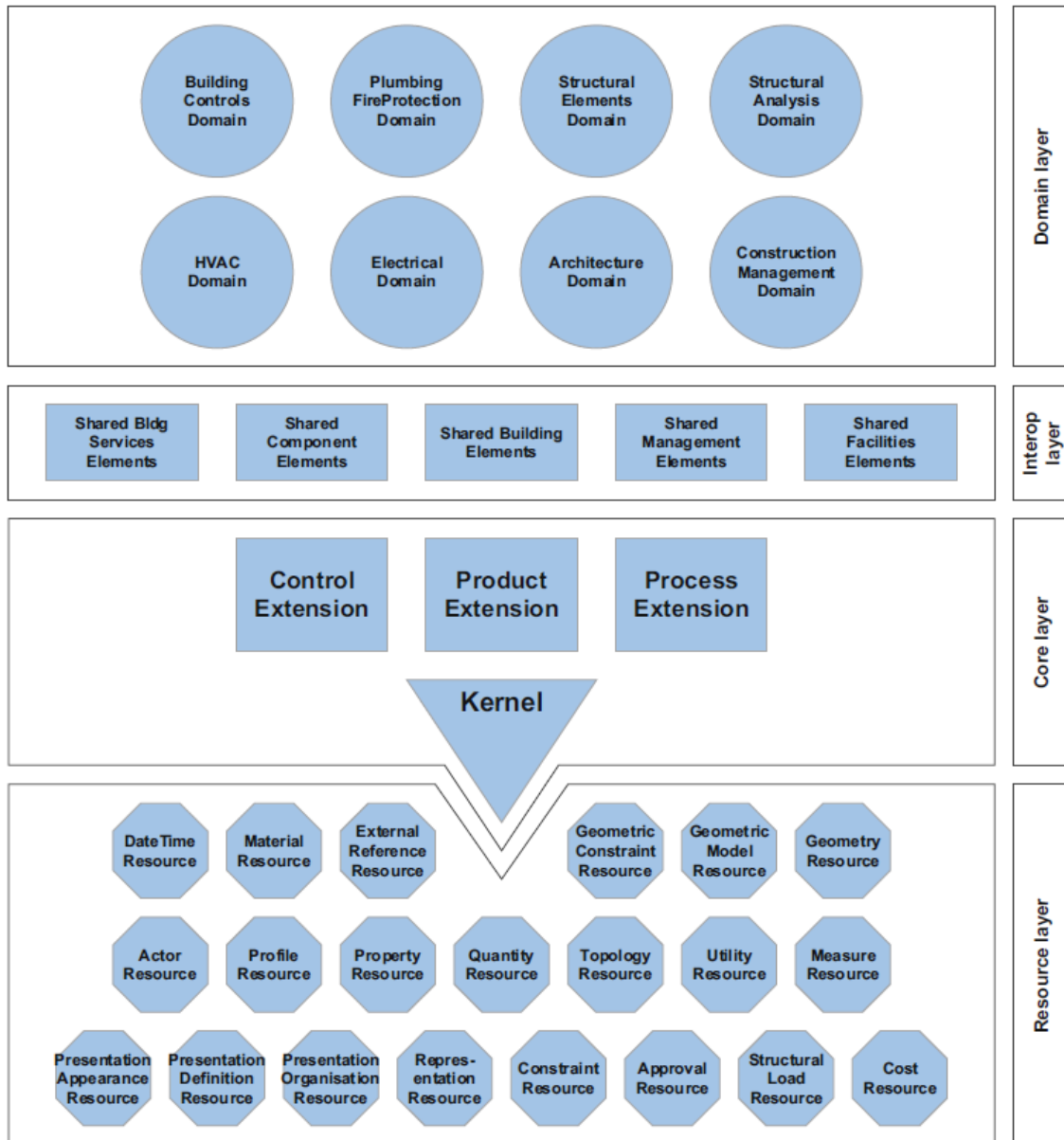
```

Figur 31 XML-data over vægtypen

Ud fra analyse af XML-data for vægtypen, kunne datamodellen oversættes til EXPRESS. Vægtype hedder IfcWallType, Type Properties hedder IfcPropertySet, og Type parameter hedder IfcPropertySingle Value.

2.3.1 Industry Foundation Class (IFC)

Der analyseres på Ifc-datamodellen med henblik på at gøre brug af generaliseringskonceptet i Ifc. Afsnittet er udarbejdet med udgangspunkt i Ifc-specifikationen. (buildingSmart, 2018)



Organization i layer

IFC-datamodellen er kompleks og omfattende, for at overskueliggør arbejdet med datamodellen, er datamodellen blevet opdelt i adskillige layer. Det overordnede princip bag de forskellige layer er, at elementerne i de øvre dele i modellen kan referere til elementer under dem, men ikke den anden vej rundt. På denne måde sikrer man at kerne-layer forholder sig uafhængig.

Core Layer

Kernelageret indeholder de mest elementære klasser i data modellen. Lagerne for oven kan referere til kernelageret. Klasserne i kernelageret definerer de basale strukturer, nøgle relationer

og generale koncepter, som så kan blive defineret mere præcist af klasserne som tilhører de ovenstående lager. Kernal schema repræsenterer kernen i IFC-datamodellen og består af abstrakte klasser såsom IfcRoot, IfcObject, IfcActor, IfcProcess, IfcProduct, IfcProject, IfcRelationship.

Kernel schemaet er udvidet af følgende tre skema: Product Extension, Process Extension and Control Extension, som også er en del af Core Layer.

Product Extension skemaet beskriver de fysiske og rummelige objekter i en bygning og deres tilhørende relationer. Product Extension dækker over sub-klasserne til IfcProduct, herunder IfcBuilding, IfcBuildingStorey, IfcSpace, IfcElement, IfcBuilding Element, IfcOpeningElement og relationsklasserne IfcRelAssociatesMaterial, IfcRelFillsElement og IfcRelVoidsElement.

Process Extension skemaet rummer klasser til beskrivelse af processer og operationer. Og giver de basale midler til definering af relationer mellem proceselementer og ressource-elementer.

Control Extension definerer klasser til kontrol af objekter, herunder IfcControl og IfcPerformanceHistory, såvel som muligheden for at tilknytte disse objekter til fysiske og rummelige objekter.

Interoperability Layer

Denne kaldes også Shared Layer, og er placeret mellem Core Layer for neden og Domain Layer for oven. Interoperability Layer er således bindeledet mellem de to, og rummer klasser for vigtige bygningselementer såsom IfcWall, IfcColumn, IfcBeam, IfcPlate, IfcWindow.

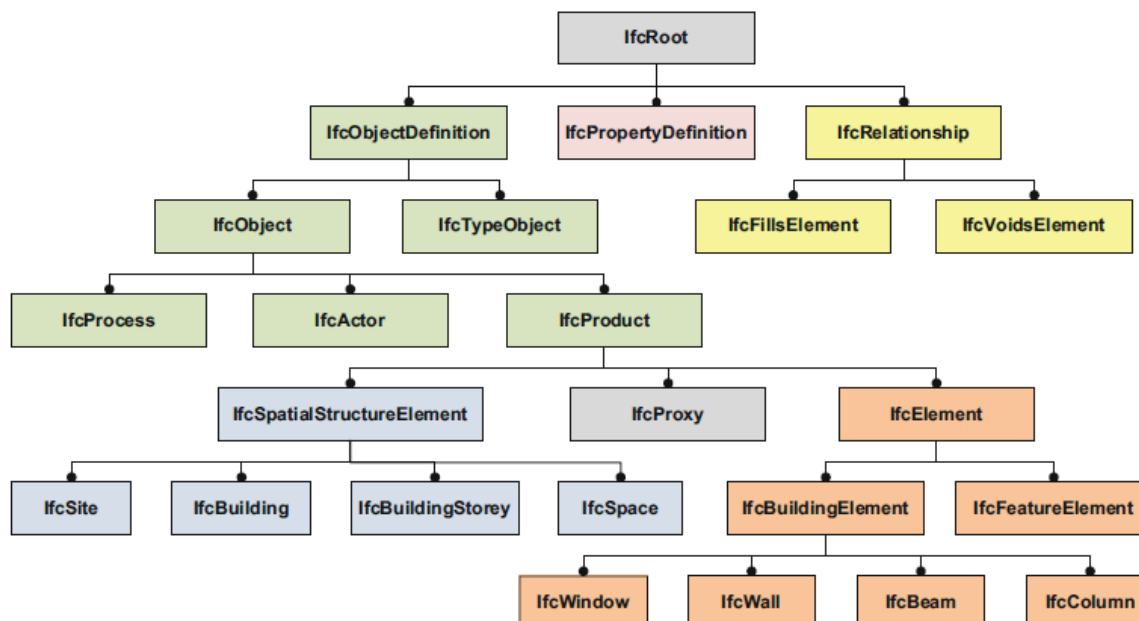
Domain Layer

Denne scheme-lager indeholder specificer klasser for objekter som bruges inden for et givent domæne eller fagområde, herunder arkitektur, bygningskontrol, udførelse, el-, vvs og brandsikkerhed.

Ressource Layer

På det nederste niveau er datastrukturer defineret som kan bruges af øvrige og samtlige andre layer. Klasserne herunder stammer ikke direkte fra IfcRoot og har ikke en

Inheritance



Figur 32 Bormann et. Al. 2018

Som i enhver anden Object-oriented data model, så spiller oprindelse eller afstamning også en væsentlig rolle i IFC-datamodellen. Figuren for oven illustrerer delelement af hierarkiske afstamning i ifc. Den hierarkiske afstamning har en logisk sammenhæng, og moduleres på baggrund af objekternes relationer til hinanden. Figuren for oven er illustrere dog kun de vigtigste relationer i IFC-modellen.

Ifc-root og dens direkte sub-classes

Alle relationer i IFC stammer fra IfcRoot. Alle objekter, klasser eller enheder, hvilket alle er synonymer for hinanden, skal derfor have en direkte eller indirekte tilknytning til IfcRoot, foruden klasserne i Ressource Layer. IfcRoot har den funktion at den tilknytter et unik ID til enhver objekt, igennem et såkaldt Globally Unique Identifier (GUID). Igennem denne Guid kan man så spore alle informationer til det pågældende objekt, herunder ejerskab, historie og oprindelse m.m. Derudover har kan alle objekter tilknyttes et navn og beskrivelse.

IfcObjectDefination, IfcPropertyDefinition og IfcRelationships afstammer direkte fra IfcRoot, dermed den næste kæde i hierarkiet.

IfcObjectDefinition er en abstrakt super-klasse som indeholder alle klasser der repræsenterer fysiske objekter for eks. bygningsdele, rumlige objekter for eks. arealer og åbninger, eller konceptuelle objekter som processer, kostspris. IfcObjectDefination rummer yderligere tre klasser IfcObject (enkelte objekter i byggeprojektet), IfcTypeObject (objekttype) og IfcContext (generelle projekt informationer).

IfcRelationship og dens sub-klasser beskriver relationer i form af objekter. Denne koncept beskrives yderligere i det efterfølgende afsnit.

IfcPropertyDefinition definerer egenskaber til et objekt, som ikke allerede er en del i IFC-datamodellen. (Bormann et. Al. 2018)

IfcProduct og subklasser

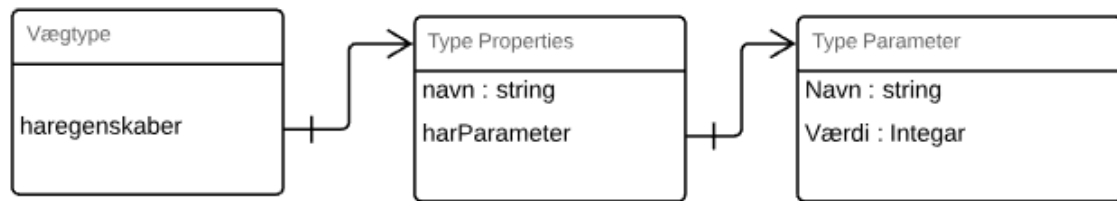
IfcProduct er en abstrakt repræsentation af alle objekter som relaterer sig til geometri eller rumlige forhold. Alle klasser som bruges til at beskrive en virtual bygning er subklasser til IfcProduct. IfcProduct objekter kan tilknyttes geometri og placering.

IfcElement er en sub-klasse til IfcProduct, men en superklasse til samtlige bygningsdele, herunder IfcWall, IfcColumn, IfcWindow etc.

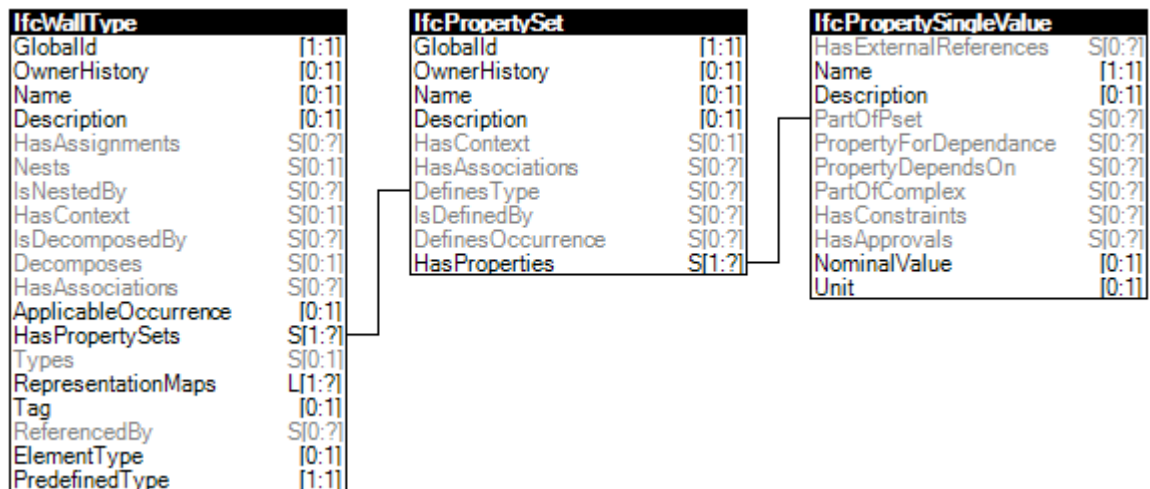
IfcSpatialElement er superklasse som dækker over alle ikke-fysiske objekter, herunder subklasserne IfcSite, IfcBuilding, IfcBuildingStorey og IfcSpace.

IfcProduct indeholder sub-klassen IfcProxy og serverer som en buffer, hvor man kan placere objekter som ikke tilhører under forhennævnte klasser. IfcProduct har subklasserne IfcAnnotation, IfcGrid og IfcPort. (Bormann, et al. 2018)

Delkonklusion på Realisering i EXPRESS

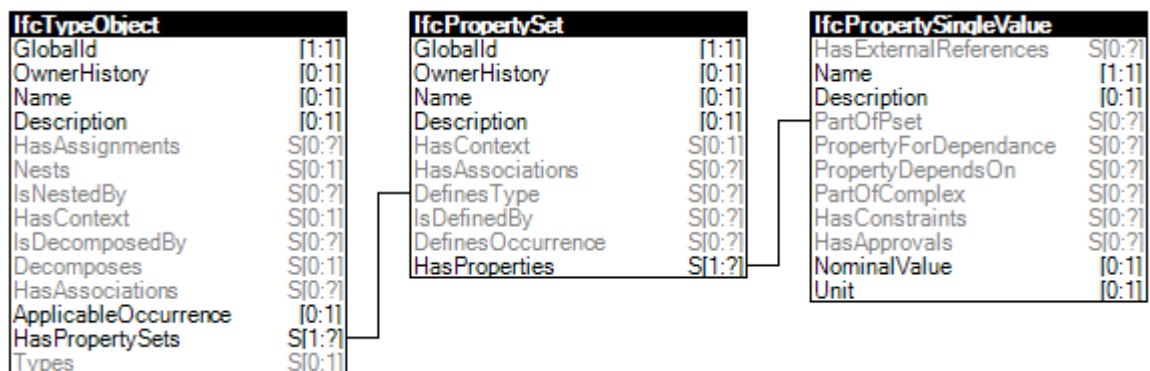
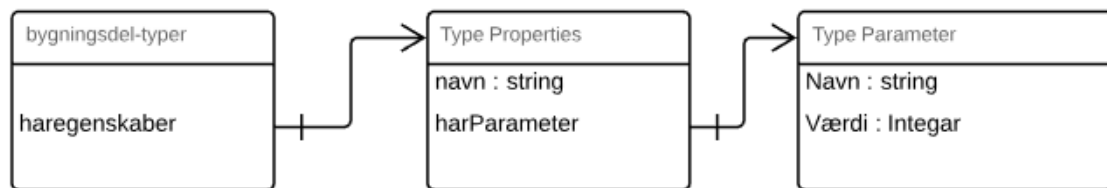


Figur 33 UML-koncept model



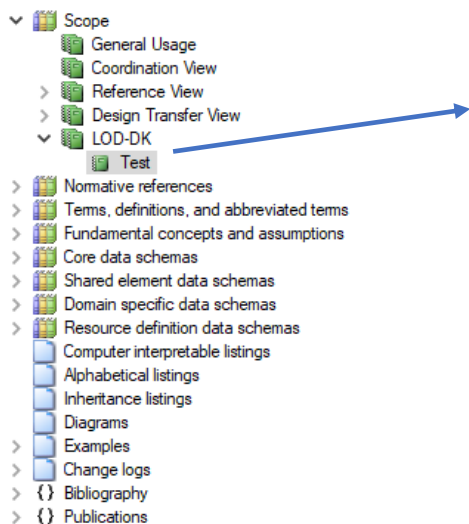
Figur 34 Realisering I Express igennem MVD-udvikling på baggrund af datmodellen

Ved at studere IFC-specifikationen kunne vi anvende generaliseringskonceptet heraf.



Konklusion på analyserende del

Baseret på trinene, som blev belyst i foregående kapitel, udvikledes en Model View Definition på baggrund af attributterne ved de enkelte egenskabsdata.



Figur 35 Kravspecifikation til attributter igennem model view definition. Produktet af analysen.

	Type navn	Width [vaeg]	Thickness [lag & deak]	Placering - Etage	Hulmål - Højde, bredde, flugtvei, opbygning	Konstruktionsopbygning
IfcDoor	X					
IfcRoof	X					
IfcStair	X					
IfcWall	X					
IfcWindow	X					
IfcSlab	X					
IfcSpace	X					
IfcWallType	X	X				
IfcDoorType	X					
IfcWindowType	X					
IfcRoofType	X					

III. Vurderende

buildingSMART introducerede IfcDoc-værktøjet til udvikling af Model View Definition (MVD). Igennem værktøjet kan en delmængde af Ifc-skemaet defineres i henhold til et konkret informationsbehov. Værktøjet har en integreret valideringsfunktion, som tillader import af en bygningsmodel i Ifc-format, hvorefter den automatisk validerer den importerede IFC-model i henhold til kravspecifikationen i et defineret MVD.⁸

Den analyserende del udviklede en Model View Definition på baggrund af en række attributter, som skal være til stede i en bygningsmodel for at overholde et vis LOD-DK-niveau. Den analyserende del lagde fundamentet for en vurdering af IfcDoc i forhold til effektivisering af valideringsprocessen. Denne del vurderer den integrerede valideringsfunktion, forud for en konklusion på rapportens problemformulering.

Hvordan kan udvikling af Model View Definition på baggrund af bygningsdelsspecifikationer effektivisere valideringsprocessen?

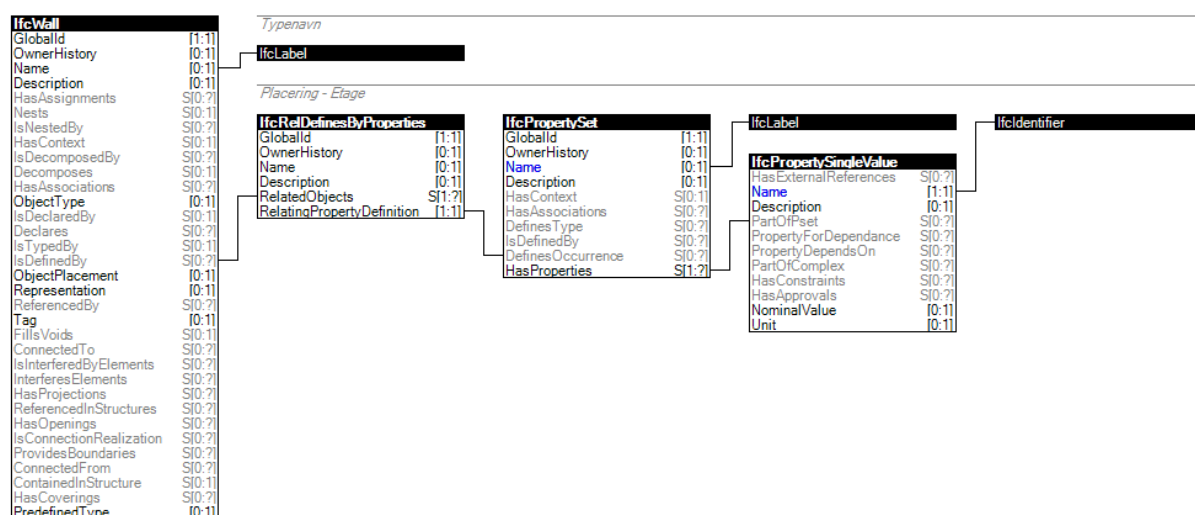
⁸ <http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool>
<http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool/ifcdoc-help-page>

3.1 Validering af Egenskabsdata Type A, B og C

I nærværende afsnit illustreres valideringsfunktionen i forhold til de enkelte egenskabsdata typer (jf. afsnit 2.1).

Væg	Type A	Type B		Type C
Egenskabsdata	Placering	Typenavn	Tykkelse	Konstruktionsopbygning
Egenskabstype	Constraints	Type Properties	Construction	Construction
Egenskabsparameter	Base Constraints	Type	Width	Structure

3.1.1 Væg: egenskabsdata Type A

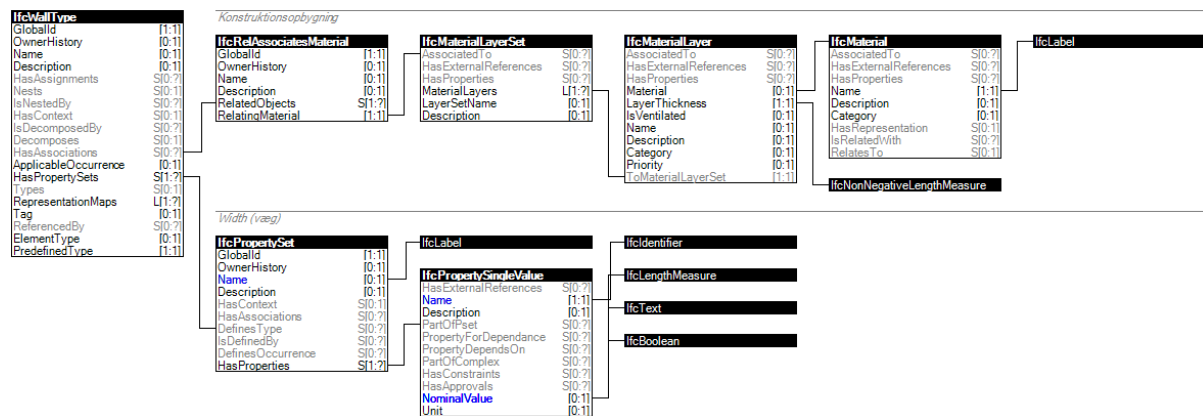


Figur 36 kravspecifikation til attributter for egenskabsdata Type A

Validering af type A

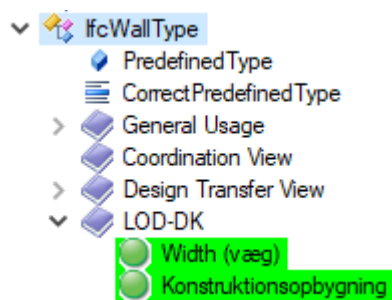
1123	IfcWall	Basic Wall:Ydervæg:343333
1365	IfcWall	Basic Wall:Ydervæg:343393
1435	IfcWall	Basic Wall:Ydervæg:343483
1505	IfcWall	Basic Wall:Ydervæg:343545
6653	IfcWall	Basic Wall:Generic - 300mm:344765
6798	IfcWall	Basic Wall:Generic - 300mm:344981
6868	IfcWall	Basic Wall:Generic - 300mm:345257
6938	IfcWall	Basic Wall:Generic - 300mm:345420
8474	IfcWall	Basic Wall:Skillevæg:346257
8600	IfcWall	Basic Wall:Skillevæg:346316
9602	IfcWall	Basic Wall:Indervæg - Bærende:347358
9747	IfcWall	Basic Wall:Indervæg - Bærende:347488

3.1.2 Egenskabsdata Type B & C



	Egenskabsparameter	Værdi	Egenskabstype	Description
▶	Width	IfcLengthMeasure	Construction	Grønt lys = Vægge har tykkelse
*				

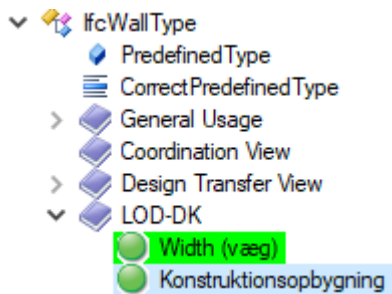
Validering af Tykkelse



1197	IfcWall...	Basic Wall:Ydervæg
6672	IfcWall...	Basic Wall:Generic - 300mm
8486	IfcWall...	Basic Wall:Skillevæg
9614	IfcWall...	Basic Wall:Indervæg - Bærende

	Egenskabsparameter	Værdi	Egenskabstype	Description
▶	Width	IfcLengthMeasure	Construction	Grønt lys = Vægge har tykkelse
*				

Validering af konstruktionsopbygning



1197	IfcWall...	Basic Wall:Ydervæg
6672	IfcWall...	Basic Wall:Generic - 300mm
8486	IfcWall...	Basic Wall:Skillevæg
9614	IfcWall...	Basic Wall:Indervæg - Bærende

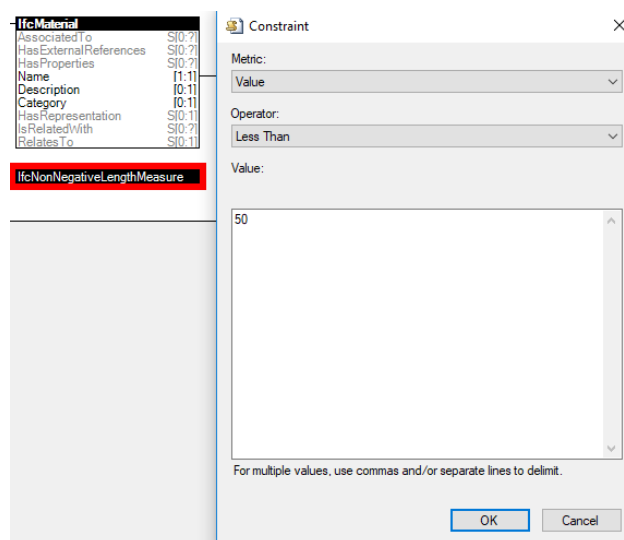
3.2 Diskussion & Resultater

3.2.1 Regelsæt

I den opstillede bygningsmodel, er væggene moduleret i henhold til LOD 400. I dette afsnit testes MVD-kravspecifikation ift. opsætning af regelsæt. Herunder sættes en regelsæt op, der siger at konstruktionslagene ikke må overstige 50 mm.

Tabel 8 Konstruktionslag i væggen

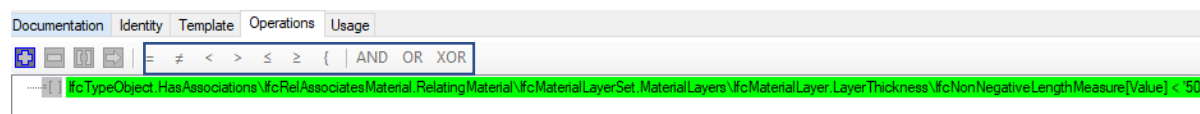
Typenavn	Konstruktionsopbygning
Ydervæg	Beton: 100 mm Isolering: 200 mm Gipsplade: 12 mm
Indervæg - Bærende	Gipsplade: 12 mm Beton: 100 Gipsplade: 12 mm
Skillevæg	Gipsplade: 12 mm Beton: 64 Gipsplade: 12 mm
Generic	300 mm



Figur 38 Max-regelsæt i IfcDoc

1596	IfcWallType	Basic Wall: Ydervæg
7057	IfcWallType	Basic Wall: Generic - 300mm
8863	IfcWallType	Basic Wall: Skillevæg
9991	IfcWallType	Basic Wall: Indervæg - Bærende

Figur 37 Valideringsresultater: regelsæt



IfcDoc har adskillige regelsæt funktioner, disse er markeret med blå i ovenstående figur. I dette tilfælde anvendtes en max. regelsæt. På baggrund af adskillige testresultater kunne det konstateres, at regelsæt funktionen er fejlbehæftet.

Valideringsresultaterne fra testen viser, at Vægtyperne skillevæg og indervæg – opfylder kravene til, at konstruktionslagene ikke overstiger 50 mm. Det er en fejl, da det fremgår af tabellen, at ingen af vægtyperne bør bestå regelsættet.

Det anskues, at grunden til at den melder grønt ved de to vægtyper, ligger i, at regelsættet kun tager det øverste lag i betragtning. Altså gipsplade 12 mm.

Family:	Basic Wall		
Type:	Skillevæg		
Total thickness:	88.0	Sample Height:	6000.0
Resistance (R):	0.0981 (m ² ·K)/W		
Thermal Mass:	11.04 kJ/K		

Layers					
EXTERIOR SIDE					
	Function	Material	Thickness	Wraps	Structural Material
1	Core Boundary	Layers Above Wrap	0.0		
2	Structure [1]	Beton	64.0	☐	☒
3	Finish 2 [5]	Gipsplade	12.0	☒	☐
4	Core Boundary	Layers Below Wrap	0.0		
5	Finish 2 [5]	Gipsplade	12.0	☒	☐

Figur 39 flyttede på konstruktionslag i skillevæg

1197	lfcWallType	Basic Wall:Ydervæg
6672	lfcWallType	Basic Wall:Generic - 300mm
8486	lfcWallType	Basic Wall:Skillevæg
9616	lfcWallType	Basic Wall:Indervæg - Bærende

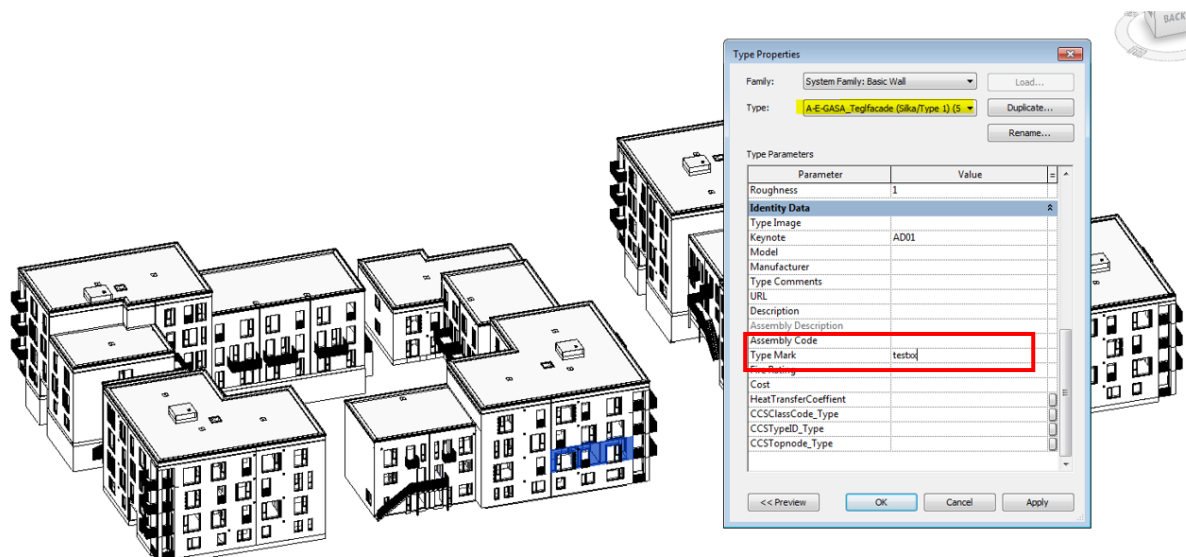
Figur 40 Bekræftelse på fejl i regelsæt-funktionen

For at bekræfte teorien, at regelsættene kun tager det øverste lag i betragtning, blev gips pladen flyttede bagerst i vægtypen ”Skillevæg”. Testresultaterne viste Rødt ved den samme skillevæg, som bestod testen før, og melder stadig grøn for den ene vægtype hvor gips pladen er placeret øverst.

På baggrund af ovenstående anskues det, at regelsæt funktionerne er fejlbehæftede. Der skal dog tages forbehold for, at funktionerne måske ikke blev brugt korrekt.

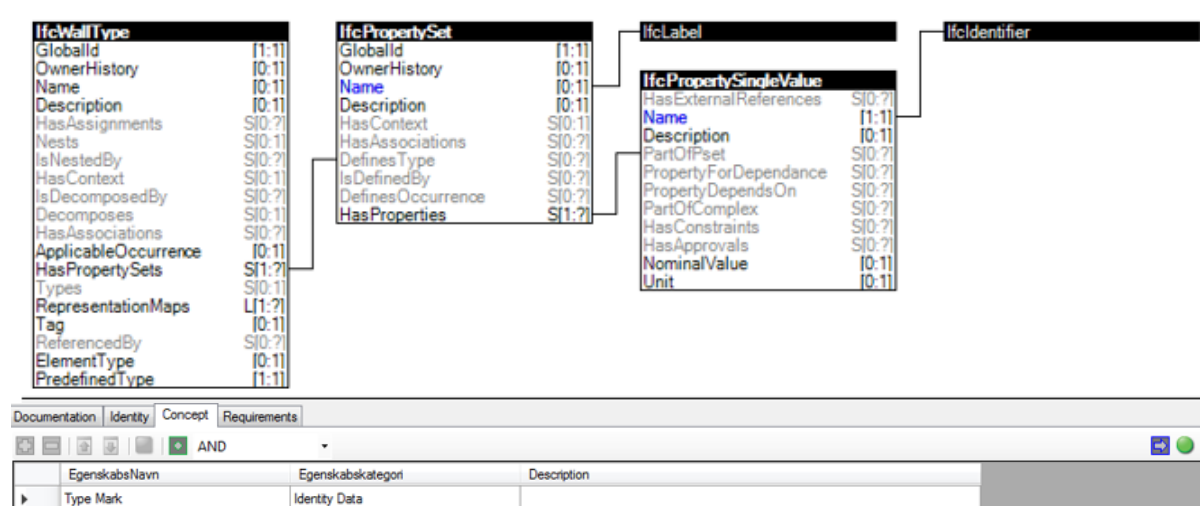
3.2.2 Test af MVD i forhold til større bygningsmodeller

For at vurdere om løsningsforslaget kan bruges i forbindelse med en større bygningsmodel, er der foretaget følgende test.



Figur 41 Test af løsningsforslaget i forhold til en større bygningsmodel

Først blev værdien ”testxx” tilføjet til en vægtype i en større bygningsmodel. Testxx indgår under parameteret Type Mark, som er en del af Identity Data. Herefter blev modellen eksporteret i et Ifc-format til validering i henhold til nedenstående kravspecifikation.



Resultatet af valideringsprocessen ses til højre. Valideringsresultaterne melder grønt ved den ene vægtype, som har en værdi ved parameteret Type Mark. Øvrige vægtyper, som ikke har en udfyldt værdi, meldes rødt.

På baggrund af testen, kan det konkluderes, at kravspecifikation til attributter igennem MVD kan resultere i en automatiseret validering uafhængig af modellens størrelse.

Det skal dog siges, at Ifc-eksport af den noget større bygningsmodel tog væsentlig længere tid.

Valideringsprocessen i IfcDoc, gik til gengæld forholdsvis hurtigt.

118.6789		
#	Type	Name
102645	IfcWallType	Basic Wall A-E-GASA_Teglfacade m. kopper ud (Type 2/...
103084	IfcWallType	Basic Wall A-E-GASA_Teglfacade (Ytong/Type 1) (504)
103826	IfcWallType	Basic Wall A-E-GASA_Teglfacade (Silka/Type 1) (504)
106341	IfcWallType	Basic Wall A-E-GASA_Teglfacade m. kopper ud (Type 2/...
131073	IfcWallType	Basic Wall A-X-GASA_Kalksandsten_Silka 1900 kg/m3 (...)
173077	IfcWallType	Basic Wall A-E-GASA_Teglfacade Indgangsparti (Silka/T...
181771	IfcWallType	Basic Wall A-E-GASA_Kældervæg (510)
183534	IfcWallType	Basic Wall A - Skillevæg 50 mm
186466	IfcWallType	Basic Wall A-E-GASA_Tegloverlægger
242168	IfcWallType	Basic Wall A-X-GASA_Porebeton_(100)
582350	IfcWallType	Basic Wall_A-I-Sketch-25 (Ståltrappevange)
621463	IfcWallType	Basic Wall A-E-GASA_Tegl hjømsøjle (528)
635056	IfcWallType	Basic Wall A-X-Stålsøjle til udv. trapper
764538	IfcWallType	Basic Wall A-X-GASA_Kalksandsten_(200)
767365	IfcWallType	Basic Wall A-X-GASA_Kalksandsten_Silka 2500 kg/m3 (...)
900686	IfcWallType	Basic Wall A-E-GASA_Kældervæg v. trappe (504) IKKE ...
905830	IfcWallType	Basic Wall A-E_Elevatorgube (420)
926029	IfcWallType	Basic Wall A-E-GASA_Trappevæg (200)
926291	IfcWallType	Basic Wall A-X-GASA_Porebeton_(150)
948196	IfcWallType	Basic Wall A-E-GASA_Teglfacade højkant (504)
976974	IfcWallType	Basic Wall_Sokkel
1873...	IfcWallType	Basic Wall_Sokkel (Ieca+Iso)
2053...	IfcWallType	Basic Wall A-E-GASA_Trælister v. indgange (88)
2056...	IfcWallType	Basic Wall A-E-GASA_Teglsten Type 1 (108)
2077...	IfcWallType	Basic Wall A-E-GASA_Tagsokkel under ydervæg (346)
2256...	IfcWallType	Basic Wall_vægfliser badeværelse
2365...	IfcWallType	Basic Wall_Elevatorvæg m. iso
2581...	IfcWallType	Basic Wall_vægfliser badeværelse_1

Figur 42 Resultat af validering i IfcDoc

3.2.3 Fordele og Ulemper ved kravspecifikation igennem MVD

I forbindelse med afprøvning af adskillige MVD-løsninger i IfcDoc, blev adskillige fordele og en række ulemper konstateret. I det nedenstående fremhæves disse.

Generalisering af typenavn

Ifc bygger på objektorienteret datamodelleringsprincipper. En af fordelene ved kravspecifikation igennem MVD er, at man ikke behøver at specificerer krav til hver enkelt bygningsdel. For eksempel, fremgår det i bygningsdelsspecifikationerne at Væg, Vindue, Tag og resterende bygningsdele skal have egenskabsdata Typenavn. Hvorimod i MVD kan man blot kravstille til typenavn, ved at lave et koncept som gælder for samtlige bygningsdele, hvori typenavn skal indgå. Denne illustreres i nedenstående figur.

The screenshot shows the IfcDoc interface. On the left, a tree view shows the project structure. The 'Fundamental concepts and assumptions' section is expanded, showing the 'Typenavn' concept. The 'Typenavn' concept is defined with properties: Width (væg), Thickness (tag & dæk), Placement - Etage, Height, breadth, height, and construction. The 'Typenavn' concept is also used to define the 'Typenavn' property for various building elements like Wall, Window, Roof, Slab, and Space. On the right, a table shows the list of building elements and their associated 'Typenavn' values.

#	Type	Name
104	IfcSpace	1
435	IfcSpace	2
592	IfcSpace	3
935	IfcSpace	4
1123	IfcWall	Basic Wall Ydervæg 343333
1365	IfcWall	Basic Wall Ydervæg 343393
1435	IfcWall	Basic Wall Ydervæg 343483
1505	IfcWall	Basic Wall Ydervæg 343545
1680	IfcDoor	M_Single-Flush Indgang 343559
4419	IfcWindow	M_Window-Casement-Double Almin...
5267	IfcWindow	M_Window-Casement-Double Almin...
5588	IfcWindow	M_Window-Casement-Double Almin...
5910	IfcWindow	M_Window-Casement-Double Almin...
6226	IfcRoof	Basic Roof-Tag 344101
6269	IfcSlab	Basic Roof-Tag 344101
6302	IfcSlab	Basic Roof-Tag 344101
6339	IfcSlab	Basic Roof-Tag 344101
6372	IfcSlab	Basic Roof-Tag 344101
6653	IfcWall	Basic Wall Generic - 300mm 344765
6798	IfcWall	Basic Wall Generic - 300mm 344901
6868	IfcWall	Basic Wall Generic - 300mm 345257
6938	IfcWall	Basic Wall Generic - 300mm 345420
6993	IfcWindow	M_Window-Casement-Double Almin...
7319	IfcWindow	M_Window-Casement-Double Almin...
7640	IfcWindow	M_Window-Casement-Double Almin...

Figur 44 brug af inheritance konceptet

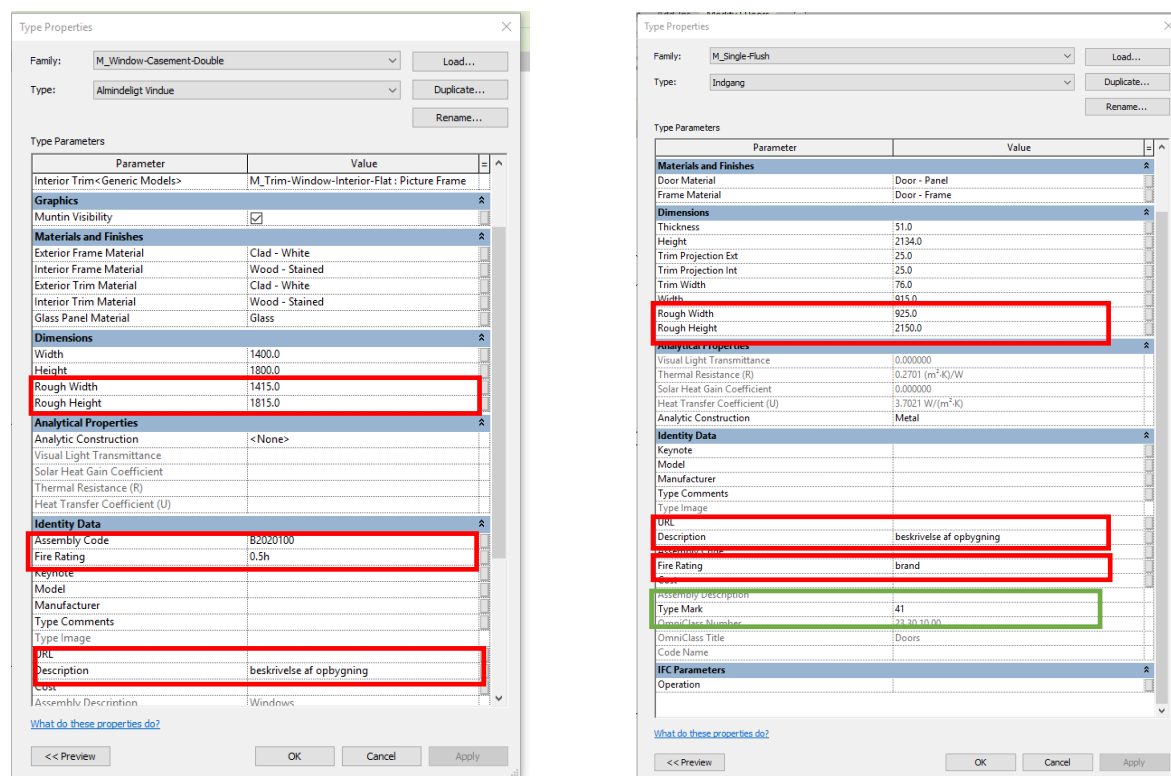
Figuren for oven viser, hvordan man kan bruge generaliseringskonceptet til at stille krav til en super-klasser. Figuren til højre viser de sub-klasser, som konceptet er gældende for. På denne måde, automatiseres validering af typenavn for samtlige gældende bygningsdele.

Model View	Entity
LOD-DK	IfcDoor
LOD-DK	IfcRoof
LOD-DK	IfcStair
LOD-DK	IfcWall
LOD-DK	IfcWindow
LOD-DK	IfcSlab
LOD-DK	IfcSpace

Figur 43 Liste over bygningsdele, som er underlagt konceptet

Generalisering: Dør og vindue LOD-DK 325/400

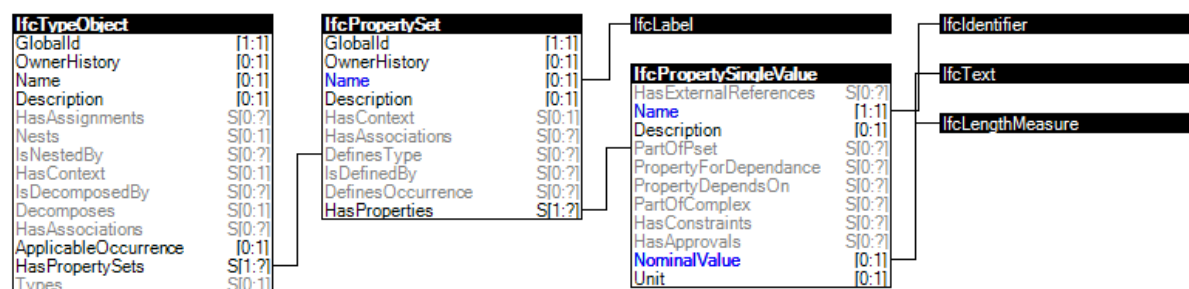
En anden fordel ved generalisering-konceptet beskrives med udgangspunkt i vindue og dør LOD 325/400. Deres egenskabsdata fremgår af nedenstående figur.



Figur 45 Egenskabsdata for vindue (højre) og dør (venstre)

Der kan drages paralleller mellem egenskabsdata for dørtyper og vindue-typer. Begge egenskabsdata findes under Revit-vinduet: Type Properties. Herudover, er de placeret næsten ens. Det antages, at Rough Width svarer til egenskabsdata: Bredde (Hulmål). Det fremgår af figuren for oven, at Rough Width er placeret under Dimensions, både for dør og vindue. De øvrige egenskabsdata, som er placeret samme sted i begge tilfælde er markeret med rødt.

Der er dog én forskel mellem krav til egenskabsdata for dør og vinduer. Det kommer i form af egenskabsdata Beslagsæt. Denne er gældende for dør, men ikke vinduer. Det antages, at egenskabsdatasæt for beslagsæt indgår under Type Mark (markeret med grønt). Generaliseringskonceptet kan bruges til kravspecifikation af de to bygningsdele på trods af deres forskelligheder.



Figur 46 Kravspecifikation til Vindue og Dør LOD 325/400 (komplettering)

Model View	Entity
LOD-DK	IfcDoorType
LOD-DK	IfcWindowType

ver af disse Entities, er specialiseret i henhold til

	Egenskabstype	Description
	Dimensions	Hulmål - Bredde
	Dimensions	Hulmål - Højde
	Identity Data	Flugtvej
	Identity Data	Opbygning

ndue

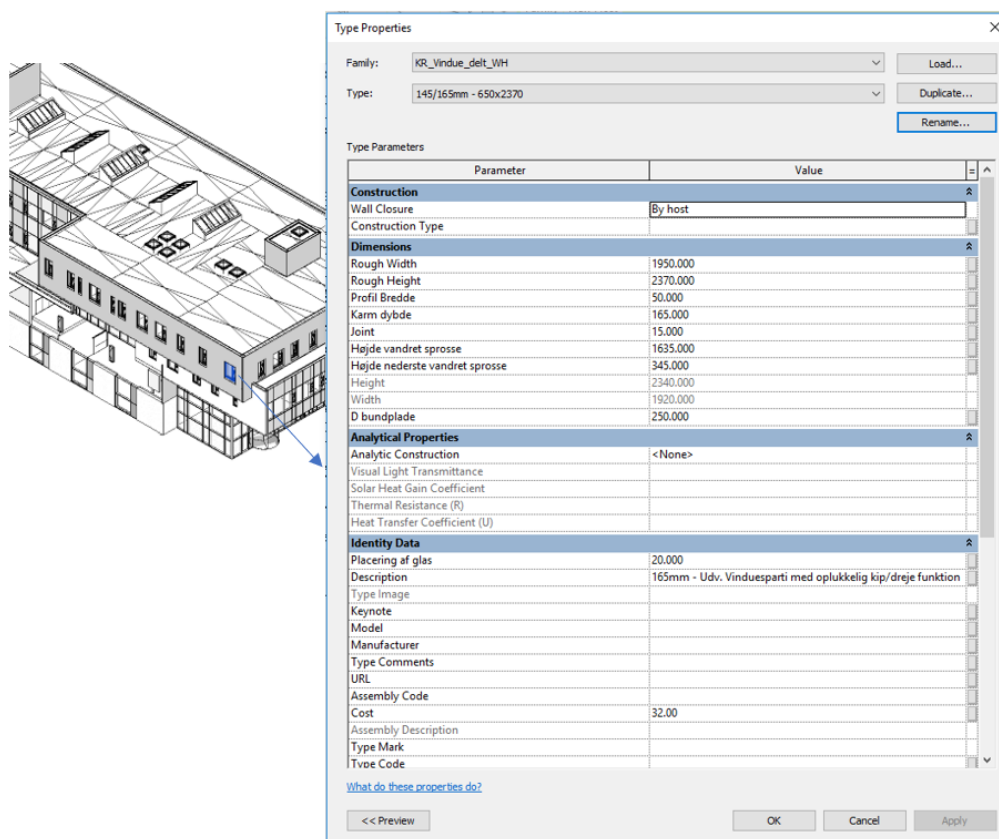
	Egenskabstype	Description
	Dimensions	Bredde - Hulmål
	Dimensions	Højde - Hulmål
	Identity Data	Flugtvej
	Identity Data	Opbygning
	Identity Data	Beslagsæt

 ∂r

ingskonceptet. I stedet for at definere en hel ny
have en "instance" af et eksisterende koncept.

Test af generaliserings-konceptet

Ovenstående koncept illustreres med udgangspunkt i en vindue-type i bygningsmodellen for TMV⁹. Det fremgår af Type Properties for vinduet, at alle egenskabsdata foruden Flugt Vej er til stede i typen. Det antages, at Flugt Vej indgår under Identity Data, under parameternavnet Fire Rating. Modellen blev eksporteret til validering iht. et defineret MVD på baggrund af egenskabsdata for vindue LOD 325/400.



Figur 50 Type Properties for vindue TMV

Documentation Identity Concept Requirements				
AND				
	Egenskabsparameter	Datatype	Egenskabstype	Description
►	Rough Width	IfcLengthMeasure	Dimensions	Hulmål - Bredde
	Rough Height	IfcLengthMeasure	Dimensions	Hulmål - Højde
	Fire Rating	IfcText	Identity Data	Flugtvej
	Description	IfcText	Identity Data	Opbygning
*				

170238	IfcWindowType	Kp - Dreje
233095	IfcWindowType	145/165mm - 1950x2370
233318	IfcWindowType	145/165mm - 1300x2370
233519	IfcWindowType	145/165mm - 650x2370
234903	IfcWindowType	210/205mm - 630x2370 - R
235264	IfcWindowType	210/205mm - 630x2370 - R
235585	IfcWindowType	175/185mm - 1300x2800
235775	IfcWindowType	175/185mm - 650x2800
236047	IfcWindowType	175/185mm - 1950x2800
236378	IfcWindowType	175/185mm - 1300x2800
326256	IfcWindowType	1300x1200
328744	IfcWindowType	175/185mm - 650x2800
335977	IfcWindowType	1300x1200
336726	IfcWindowType	175/185mm - 1950x2800
358008	IfcWindowType	210/205mm - 1300x2370
358273	IfcWindowType	205/210mm - 650x2370
358562	IfcWindowType	210/205mm - 1950x2370
467760	IfcWindowType	Tophængt
597226	IfcWindowType	Tophængt
695390	IfcWindowType	175/185mm - 650x2800
839017	IfcWindowType	210/205mm - 1950x2370
853702	IfcWindowType	Røg trappehus
870814	IfcWindowType	175/185mm - 1300x2800
912220	IfcWindowType	Default
934839	IfcWindowType	Default
948621	IfcWindowType	Default

Figur 49 Valideringsresultater for Fire Rating.

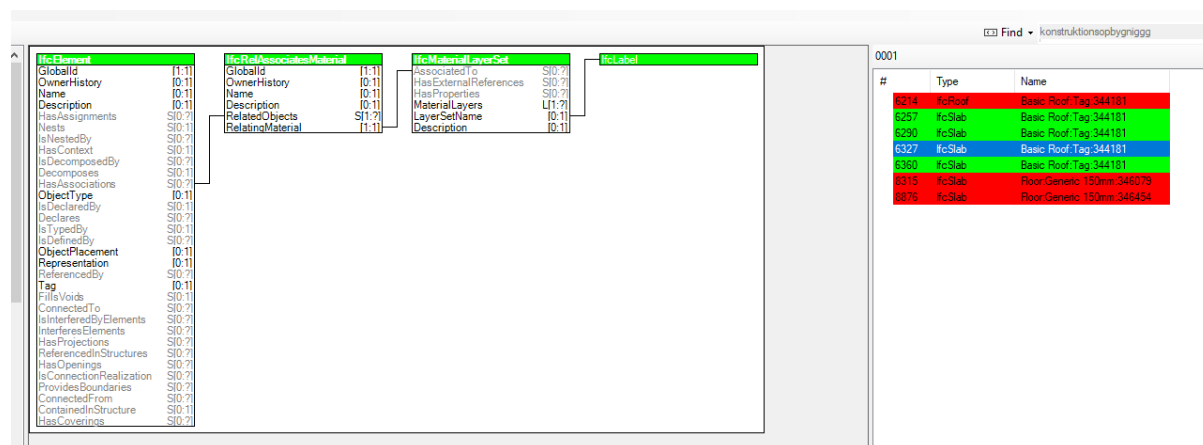
Valideringsresultaterne viste, at ingen af vindue-typerne har samtlige egenskabsdata. Hvis man trykker på de enkelte vinduestyper, fremgår det hvilke egenskabsdata er til stede, og hvilke mangler.

Figur 49 Valideringsresultater for Fire Rating.

Valideringsresultaterne viste, at ingen af vindue-typerne har samtlige egenskabsdata. Hvis man trykker på de enkelte vinduestyper, fremgår det hvilke egenskabsdata er til stede, og hvilke mangler.

⁹ Institut for byggeri og anlæg, www.byggeri.aau.dk/

Ulempe: Mangelfuld semantik



Figur 51 Roof opfattes som Slab

Revit Architecture kategoriserer Tag under Slab. Det skyldes mangel på semantisk uniformitet.

Dette problem blev også beskrevet i en af litteraturstudierne:

Most of these problems can be traced to the lack of semantic uniformity in the way BIM tools map their internal objects to IFC entities and properties. (Venugopal et al., 2012)

Ulempe: IfcDoc er stadig under udvikling

En anden væsentlig ulempe er, at værktøjet IfcDoc er i beta udgave. Det medfører, at der forekommer fejl og mangler i programmet. Desuden er det svært at finde brugsanvisninger. Herudover, kræver udviklingen af MVD i IfcDoc en dybdegående forståelse af Ifc-specifikationen.

IV. Konklusion og perspektivering

Afsnittet konkluderer på rapportens problemformulering:

Hvordan kan udvikling af Model View Definition på baggrund af bygningsdelsspecifikationer effektivisere valideringsprocessen?

Udvikling af MVD på baggrund af bygningsdelsspecifikationer (LOD-DK-niveauer) automatiserer kvalitetssikring af informationsindholdet i bygningsmodeller igennem valideringsfunktionen i IfcDoc. Den automatiserede validering er dog afhængig af placeringen af de egenskabsdata, som ligger til grund for valideringen. Denne begrænsning ved løsningsforslaget uddybes med udgangspunkt i Egenskabsdata Bredde gældende for vinduer.

Egenskabsdata for bredde findes i Revit Vinduet: Type Properties, Property Type: Dimensions, under parameteret Rough Width.

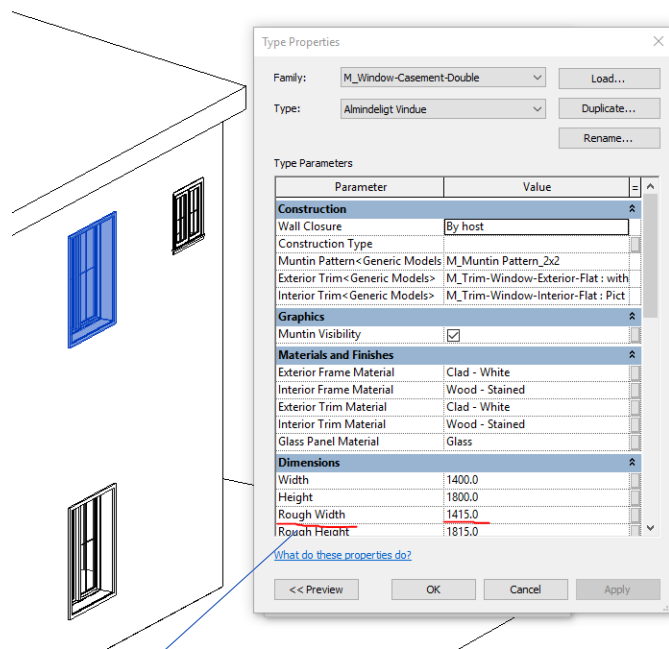
Løsningsforslaget automatiserer valideringen, så længe bredde forekommer i placeringen, som belyst i figuren til højre. Løsningsforslaget er således ikke gældende for et Ifc-Eksport fra et andet software, hvor Bredde indgår et andet sted.

En universelt automatiseret validering igennem MVD-tilgangen, kræver et tæt samarbejde mellem alle involverede parter. (udddybes i næste side)

Den besværlige valideringsproces bør takles ved roden af problemet, hvilket er en uklar kravspecifikation i forhold til indholdet i

bygningsmodeller. Udfordring ved en konkretisering på attributniveau

ligger i, at attributterne findes forskellige steder og hedder noget forskelligt afhængig af hvilket software der er tale om. Dette danner behovet for kravspecifikation igennem et neutral sprog.



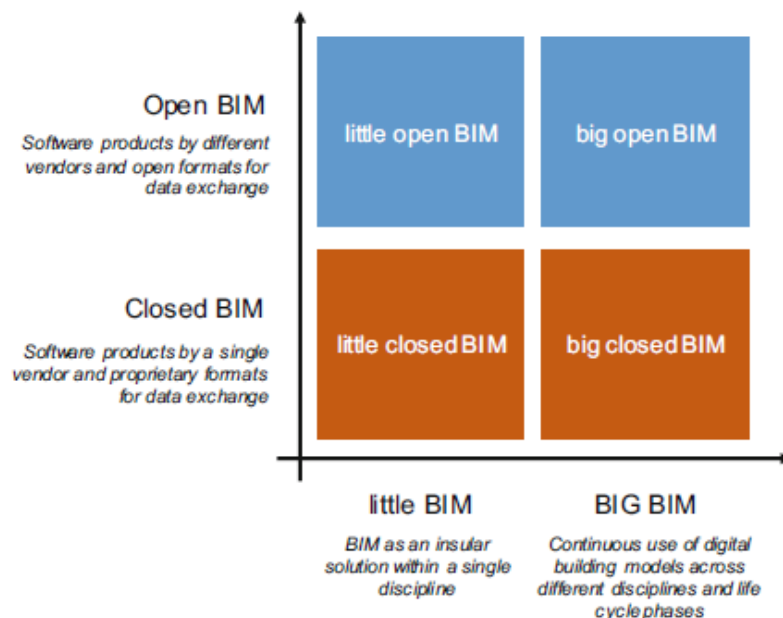
Documentation Identity Concept Requirements				
AND				
Egenskabsparameter	Datatype	Egenskabstype	Description	
Rough Width	IfcLengthMeasure	Dimensions	Hulmål - Bredde	
Rough Height	IfcLengthMeasure	Dimensions	Hulmål - Højde	
Fire Rating	IfcText	Identity Data	Flugtvej	
Description	IfcText	Identity Data	Opbygning	
*				

Figur 52 den udviklede tilgang til validering er afhængig placering af egenskabsdata

Perspektivering

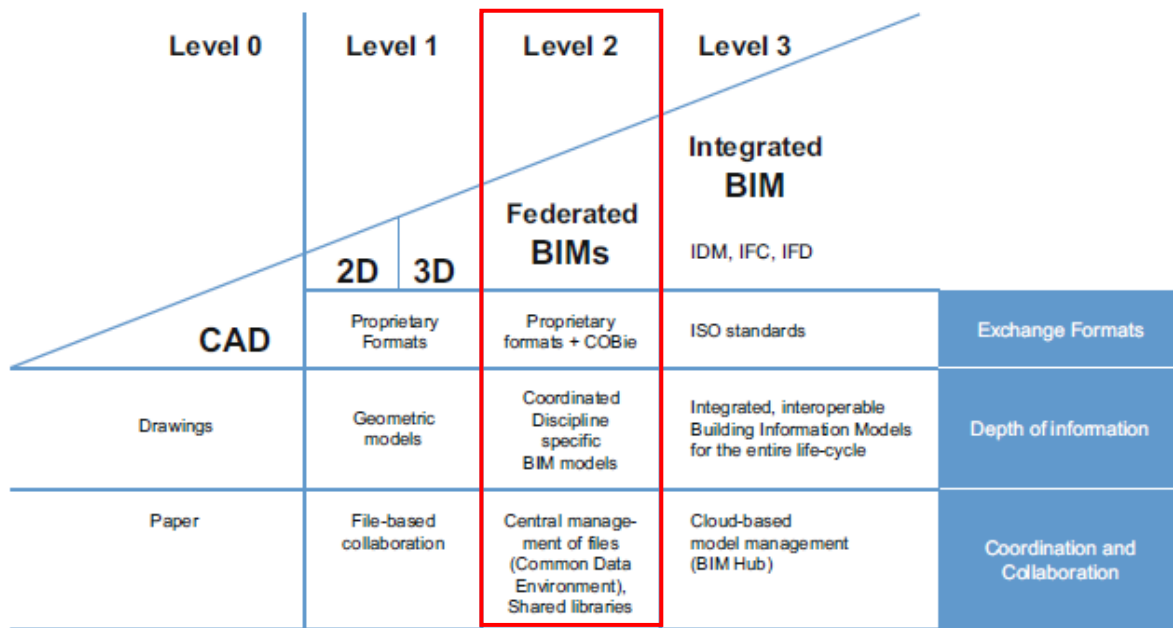
I betragtning af antallet heterogene BIM-værktøjer i byggebranchen, er der behov for kravspecifikation igennem et neutralt format. Det er her MVD træder i karakter. MVD-konceptet kræver et tæt samarbejde mellem aktørerne fra hver sin ende i værdikæden. Aktørerne skal i fællesskab definere delmængder i IFC-modellen baseret på deres enkelte informationsbehov. På baggrund af, at MVD-definitionen kræver et tæt samarbejde, har buildingSMART udviklet Information Delivery Manual (IDM). MVD/IDM går hånd i hånd. (buildingsmart 2012, 2013)

IDM/MVD-tilgangen opfanger kontekst-relateret krav igennem Information Delivery Manual og sikrer teknisk implementering af kravene igennem et Model View Definitions. Fællesmålet er at reducere rum til fortolkninger og effektivisere Ifc-baseret informationsudveksling.



Figur 53 BIG BIM, LITTLE BIM.. (Jernigan, 2008)

Open BIM beskriver BIM-implementering hvor et neutralt dataformat bruges til udveksling af data på tværs af heterogene software. Big BIM beskriver en konsistent modelbaseret kommunikation på tværs af samtlige aktører igennem byggefaserne. (Jernigan, 2008) Ifc-baseret informationsudveksling er derved nøglen til BIG-OPEN BIM.



Figur 54 Byggebranchen befinder sig i Level 2 (Bew, M., & Richards, 2011)

Indledningsvis blev det nævnt, at staten lancerede initiativet Det Digitale Byggeri ved årtusindskiftet for at modvirke den lave produktivitet i byggebranchen. Principperne for Det Digitale Byggeri svarer til en Federated BIMS. Det betyder to ting. Et: branchen er moden nok til en opgradering hvor der høstes gevinster af Integrated BIM. 2: Staten kan igangsætte denne udvikling igennem kravsstillelse til offentlige bygherre, som da man løftede branchen fra Level 1 til Level 2.

V. Bibliography

- Fisker, S. et al., 2011. *Anlægsteknik 2 - Styling af byggeprocessen*. 3. red. s.l.:Polyteknisk forlag.
- Bew, M., & Richards, M. (2011) 'BIM maturity model, strategy paper for the government construction client group'. Available at:
<https://www.cdbb.cam.ac.uk/Resources/ResourcePublications/BISBIMstrategyReport.pdf>.
- BIM7AA & DiKon (2018) 'Bygningsdelsspecifikation'.
- BIMForum (2018) *Level of development specification*. Available at: <http://bimforum.org/lod/>.
- bips (2014) *CCS Informationsniveau*. Available at:
https://bips.dk/files/news_files/informationsniveau_20141211_r0.pdf (Accessed: 15 May 2017).
- Booch, G. (2005) *Object-oriented analysis and design - Understanding System Development with UML 2.0*, NY The Benjamin. Available at:
<http://pedram5700.persiangu.com/document/0805353402.pdf>.
- buildingsmart (2018) *Industry Foundation Classes*. Available at: <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/>.
- Chuck Eastman, Paul Teicholz, Rafael Sacks, K. L. (2011) *BIM Handbook: A guide to Building Information Modeling for Owners, Managers, Engineers, and Contractors*.
- Danmarks Regering (2003) *Staten som bygherre*. Available at:
http://www.evm.dk/resources/oem/static/publikationer/html/bygherre/EBS_byggeri.pdf.
- Eastman, C. et al. (2011) *A GUIDE FOR DEVELOPMENT AND PREPARATION OF A NATIONAL BIM EXCHANGE STANDARD*. Available at:
http://dcom.arch.gatech.edu/pcibim/documents/IDM-MVD_Development_Guide_v4.pdf (Accessed: 8 September 2018).
- Fahad, M. et al. (2017) 'Towards Validation of IFC Models with IfcDoc and SWRL A Comparative Study', (c), pp. 7–13.
- Jespersen, J. M. (2008) 'Det Digitale Byggeri', *Hfb Opslag*, (Implementerings: netværket), pp. 19–24. Available at:
http://www.hfb.dk/fileadmin/templates/hfb/dokumenter/artikler/Det_digitale_byggeri.pdf.
- Kreider, M. & D. (2010) 'Determining the frequency and impact of applying BIM for different purposes on building projects', *In Proceedings of the 6th International Conference on Innovation in Architecture, Engineering and Construction*.
- Kvale, S., Brinkmann, S. and Forlag, H. R. (2009) *Introduktion til et håndværk 2. udgave*.
- König, C. K. & M. (2018) 'Data Modelling', *Building Information Modeling: Technology Foundation and Industry Practice*.
- Lee, Y.-C., Eastman, C. M. and Solihin, W. (2016) 'An ontology-based approach for developing data exchange requirements and model views of building information modeling'. doi: 10.1016/j.aei.2016.04.008.
- Licitation.dk (2019) *lindbjergskolen*. Available at:
<https://www.licitationen.dk/project/view/881/lindbjergskolen>.
- Pinheiro, S. et al. (2018) 'MVD based information exchange between BIM and building energy performance simulation'. doi: 10.1016/j.autcon.2018.02.009.
- silberschatz, korth, sudarshan (2002) 'Database system concepts 4th edition'.
- Venugopal, M. et al. (2012) 'Semantics of model views for information exchanges using the industry foundation class schema', *Advanced Engineering Informatics*, 26, pp. 411–428. doi: 10.1016/j.aei.2012.01.005.

VI. Figurliste

Figur 1 Bygherren stiller krav til øvrige parter igennem IKT-specifikationer (egen tilvirkning).....	8
Figur 2 Aftalegrundlaget blandt rådgivende og udførende (egen tilvirkning)	9
Figur 3 Udklip af leverancespecifikationer på casen (jf. bilag: leverancespecifikation).....	10
Figur 4 LOD-DK-kobling til øvrige standarder (BIM7AA & DiKon, 2018).....	11
Figur 5 Bygningsdelsspecifikation over vægge (DiKon & BIM7AA, 2018).....	12
Figur 6 Model View Definition (Venugopal et. Al., 2012).....	15
Figur 7 datamodelleringsprocessen (Bormann et. al. 2018)	17
Figur 8 Opstillet "virkelighed", egenudviklet model i Revit.....	24
Figur 9 Rum LOD 325.....	25
Figur 10 Type A: Attributterne for egenskabsdata findes under vinduet: Properties	26
Figur 11 Vægge i modellen	27
Figur 12 Egenskabsdata over vægge.....	27
Figur 13 Type B: attributterne for egenskabsdata findes under Type Properties, Type C: er markeret med grønt	28
Figur 14 Vinduer i bygningsmodellen	29
Figur 15 Egenskabsdata for vindue	30
Figur 16 Class-diagram (koch & konig, 2018)	32
Figur 17 Eksempel væg.....	33
Figur 18 Entities (objekter) som indgår i væggen.....	33
Figur 19 UML-Notation for Entity Types (Class)	33
Figur 20 forskellige datatyper.....	34
Figur 21 attributter i object-orienterede datamodeller: datamodel (venstre) og data (højre)	35
Figur 22 Relationer i Objectoriented modeling – datamodel til venstre, data til højre	36
Figur 23 Relation class i object-oriented model og dens resolution.	37
Figur 24 Aggregation.....	37
Figur 25 Composition.....	38
Figur 26 Inheritance i Object-Oriented modellering datamodel (venstre) & data (højre).....	39
Figur 27 Rekonstruering af en datamodel ud fra eksisterende data.	40
Figur 28 UML-diagram over egenskabsdata Tykkelse for vægge.....	40
Figur 29 Datamodel for Type B egenskabsdata (jf. afsnit 2.1)	41
Figur 30 UML-diagram over vægtype.....	42
Figur 31 XML-data over vægtypen	43
Figur 32 Bormann et. Al. 2018	46
Figur 33 UML-koncept model.....	48
Figur 34 Realisering I Express igennem MVD-udvikling på baggrund af datmodellen	48
Figur 35 Kravspecifikation til attributter igennem model view definition. Produktet af analysen.	49
Figur 36 kravspecifikation til attributter for egenskabsdata Type A	51
Figur 37 Valideringsresultater: regelsæt	53
Figur 38 Max-regelsæt i IfcDoc	53
Figur 39 flyttede på konstruktionslag i skillevæg.....	54
Figur 40 Bekræftelse på fejl i regelsæt-funktionen.....	54
Figur 41 Test af løsningsforslaget i forhold til en større bygningsmodel.....	55
Figur 42 Resultat af validering i IfcDoc.....	56
Figur 43 Liste over bygningsdele, som er underlagt konceptet.....	57
Figur 44 brug af inheritance konceptet	57
Figur 45 Egenskabsdata for vindue (højre) og dør (venstre).....	58
Figur 46 Kravspecifikation til Vindue og Dør LOD 325/400 (komplettering).....	58
Figur 47 Specialisering på baggrund af egenskabsdata for vindue.....	59
Figur 48 Specialisering på baggrund af egenskabsdata for dør.....	59
Figur 49 Valideringsresultater for Fire Rating.	60
Figur 50 Type Properties for vindue TMV.....	60
Figur 51 Roof opfattes som Slab	61
Figur 52 den udviklede tilgang til validering er afhængig placering af egenskabsdata	62
Figur 53BIG BIM, LITTLE BIM.. (Jernigan, 2008).....	63
Figur 54 Byggebranchen befinder sig i Level 2 (Bew, M., & Richards, 2011)	64

VIII. Bilagoversigt

Bilag 1 – Interview Frederik, BIM-koordinator, Årstiderne Arkitekter

Bilag 2 – Interview Michael Ørsted

Bilag 3 – IKT-specifikationer på LBS

Bilag 4 – IKT-Procesmanual

Bilag 5 – Bygningsdelsspecifikationer

Bilag 6 - Leverancespecfikationer