



**AALBORG
UNIVERSITET**

Datahåndtering og -udveksling mellem Building Information Models (BIM) og Life Cycle Assessment (LCA)

Kandidatspeciale i bygningsinformatik

Af

Andreas Brund Jensen

Titel:

Datahåndtering og -udveksling mellem Building Information Models (BIM) og Life Cycle Assessment (LCA)

Projekt:

Kandidatspeciale, 4. semester

Vejleder:

Kjeld Svidt

ECTS:

30

Normalsider:

50

Projektperiode:

1. september 2023 til 11. januar 2024

Afleveringsdato:

11. januar 2024

Institut:

BUILD - Institut for Byggeri, By og Miljø ved Aalborg Universitet

Forfatter og underskrift:

Andreas Brund Jensen

Studienummer: 20216582

Forord

Dette speciale er udfærdiget som et 30 ECTS kandidatspeciale ved Aalborg Universitet på byggeledelse og bygningsinformatik uddannelsen med specialisering i bygningsinformatik. Specialet er udarbejdet i perioden 1. september 2023 til 11. januar 2024.

En stor tak skal lyde til Kjeld Svidt, vejleder og lektor ved Aalborg Universitet, for sin vejledning og sit engagement igennem hele specialeperioden.

Den rådgivende arkitekt- og ingeniørvirksomhed, som indvilligede i et samarbejde, skal ligeså have stor tak. Specialet havde ikke været det samme uden. Hertil gives en særlig tak til BIM-specialisterne, LCA-specialisterne og bygningskonstruktøren fra virksomheden. Jeres sparring og entusiasme undervejs har været guld værd for specialet.

Abstract

To reduce greenhouse gas emissions from new buildings, an addition was made to the Danish building regulations. It stated that all new buildings with a heated floor area that exceeds 1,000 m², would be required to account for their greenhouse gas emissions. Furthermore, not to exceed 12 kg CO₂-equivalent per m² per year. Greenhouse gas emissions can be calculated by conducting Life Cycle Assessments (LCA), wherein a building is analysed from cradle to grave. Acquiring data from Building Information Models (BIM) is fundamental for BIM-LCA integration, as quantities from building elements are essential for LCA. The required data exchange between BIM and LCA can be facilitated by utilizing Industry Foundation Classes (IFC). However, the lack of interoperability between BIM and LCA remains a significant challenge. Therefore, this study investigates how IFC can support the development of LCA and optimize the data exchange between BIM and LCA software.

A user-centred methodology was developed, drawing inspiration from Contextual Design. Furthermore, a big engineering and architecture consultancy was selected as user. They agreed to show which processes they conducted when performing LCA, which later would be compared to the systematic literature conducted in this study. The comparison would function as the scope definition for the following investigation. Afterwards, a prototype was developed to solve the challenges regarding the data exchange between BIM and LCA software, meanwhile optimizing the user's current workflow. Python was chosen as the preferred programming language, and PyCharm as an integrated development environment. Multiple challenges arose from developing the prototype and getting feedback from the user. The most severe challenge was the missing quantities in building elements when exporting the building model to IFC. Defining a standardised method for construction take-offs when no quantity data is accessible in IFC elements is crucial for BIM-LCA integration.

The prototype was capable of extracting data from IFC models and integrating it with LCA software, as well as vice versa. Furthermore, updating data in the prototype is possible without extracting quantities from IFC elements. At its current state, the prototype is unable to compute all types of IFC elements, and this should be prioritised in future updates. A user interface to enhance useability should be included in a future update as well.

Keywords: Building Information Models (BIM), Life Cycle Assessment (LCA), Industry Foundation Classes (IFC), Life Cycle Inventory (LCI), Contextual Design (CD), quantity take-off, interoperability, LCAbyg, Python.

Indholdsfortegnelse

1. Introduktion.....	1
2. Metode	3
2.1 Indledende empiriindsamling.....	4
2.2 Systematisk litteraturreview	5
2.2.1 Bearbejdning af litteraturgrundlag.....	5
2.2.2 Sammenstilling mellem teori og praksis.....	5
2.3 Udvikling af prototype.....	6
2.3.1 Konceptdemonstration.....	6
2.3.2 Feedback og videreudvikling	6
3. Analyse.....	7
3.1 Indledende empiriindsamling.....	7
3.1.1 Dannelse af vidensgrundlag	7
3.1.2 Undersøgelse af arbejdsprocesser	7
3.2 Systematisk litteraturreview	9
3.3 Begrebsafklaringer.....	11
3.3.1 Life cycle inventory (LCI).....	11
3.3.2 Bill of Quantities (BoQ).....	11
3.3.3 Environmental Product Declaration (EPD)	11
3.3.4 Life-cycle costing (LCC)	11
3.3.5 Level of Development (LOD).....	12
3.3.6 Model View Definition (MVD)	12
3.3.7 Information Delivery Manual (IDM)	12
3.3.8 Exchange Requirements (ERs)	12
3.3.9 Application Programming Interface (API).....	13
3.4 Bearbejdning af litteraturgrundlag.....	14

3.4.1 Integrationsstrategier	14
3.4.2 Perspektivering til energiområdet.....	17
3.4.3 BIM-LCA værktøjer	18
3.4.4 anbefalinger til et fremtidigt workflow.....	23
3.5 Sammenstilling mellem teori og praksis.....	27
3.5.1 Opsummering af udfordringer	27
3.5.2 Sammenstilling og afgrænsning	28
4. Udvikling af prototype	31
4.1 Præsentation af LCAbyg	31
4.2 Udvikling af nyt workflow.....	35
4.3 Konceptdemonstration.....	39
4.4 Opfølgning på konceptdemonstration	41
4.5 Præsentation af prototype	44
5. Teknisk gennemgang af prototype.....	45
5.1 Create from IFC.....	45
5.2 Update from LCAbyg.....	48
5.3 Update from IFC	49
5.4 Create from LCI.....	51
5.5 Update from LCI.....	51
6. Feedback og videreudvikling	52
7. Diskussion	54
7.1 Metode	54
7.2 Prototypen.....	55
8. Konklusion	57
9. Litteraturliste	59

1. Introduktion

I det danske bygningsreglement blev det indført, at der fra d. 1. januar 2023 skulle stilles krav til bygningers samlede klimapåvirkning. Klimapåvirkningskravet er gældende for nybyggerier, hvor det opvarmede etageareal overstiger 1.000 m². Indførelsen af kravet sker på baggrund af den nationale strategi for bæredygtigt byggeri, der indeholder initiativer til at reducere de store mængder energi og ressourcer som forbruges på opførelse, drift og vedligehold af bygninger og bygværker. Strategien er et understøttende tiltag til den vedtagne klimalov, som "[...] lovfæster et mål om, at Danmark skal reducere sin CO₂-udledning med 70 pct. i 2030 sammenlignet med 1990." (Indenrigs- og Boligministeriet, 2021). Nybyggeriernes klimapåvirkning skal vurderes igennem LCA-beregninger (Life Cycle Assessment), hvortil en grænseværdi på 12 kg CO₂-ækvivalenter pr. m² pr. år stilles som krav. Grænseværdierne er gældende frem til år 2025, hvorefter de revideres (Indenrigs- og Boligministeriet, 2021).

Ifølge ISO-standarden DS/EN ISO 14044 (2008) består en LCA af fire trin: formål og afgrænsning, kortlægning, vurdering af miljøpåvirkninger og fortolkning. Kvaliteten af en LCA afhænger af afgrænsningen og detaljeringsniveauet, der fastlægges efter den tilsigtede anvendelse. I bygningsreglementets paragraf 297 stk. 2 er en betragtningsperiode på 50 år fra byggeriets færdigmelding sat som afgrænsning. Yderligere stilles der krav til, at bestemte moduler fra forskellige livscyklusstadier indgår i LCA-beregningen. En fuldstændig LCA indeholder 17 moduler (A1-5, B1-7, C1-4 og D) fordelt på fem livscyklusstadier: produktionsfasen, konstruktionsfasen, brugsfasen, bortskaffelsesfasen og næste produktionssystem (DS/EN ISO 15804, 2019). I tabel 1 fremgår det, hvilke moduler der påkræves i LCA-beregningen jævnfør forskrifterne i bygningsreglementet.

Livscyklusstadiet	Modul	Forklaring
Produktionsfase	A1	Råmaterialer
Produktionsfase	A2	Transport
Produktionsfase	A3	Fremstilling
Brugsfase	B4	Udskiftning (dog undtaget transport og udskiftningsproces)
Brugsfase	B6	Energiforbrug til drift
Bortskaffelsesfase	C3	Forbehandling af affald
Bortskaffelsesfase	C4	Bortskaffelse
Næste produktionssystem	D	Potentiale for genbrug, genanvendelse og anden nyttiggørelse

Tabel 1. Påkrævede moduler i LCA-beregningen jf. bygningsreglementets §297 stk. 2.

I den nationale strategi for bæredygtigt byggeri anbefales det, at "[...] tage udgangspunkt i BIM for at gøre LCA og LCC-beregninger lettere og mere fleksible at foretage, fordi BIM i forvejen vil indeholde en lang række relevante data." (Indenrigs- og Boligministeriet, 2021). Begreberne Building Information Model, Modeling og

Management kan alle forkortes som BIM. I strategien refereres der til Building Information Modeling, der af Eastman et al. (2018) beskrives som en modelleringsteknologi med tilhørende processer, der bruges til at producere, kommunikere og analysere bygningsmodeller.

En BIM-LCA integration kræver, at relevante data fra bygningsmodeller og kan anvendes til LCA-beregninger. Ifølge Wastiels og Decuypere (2019) eksisterer fem integrationsstrategier, hvor der grundlæggende skelnes mellem, hvorvidt LCA-beregninger foretages i BIM- eller LCA-software. Adskillige studier har argumenteret for, at LCA-software bør foretrækkes over BIM-software til beregningerne (Xu et al., 2019; Xu et al., 2021; Alwan & Ilhan Jones, 2022; Klumbyte et al., 2023). Det skyldes primært, at mere retvisende og præcise LCA-beregninger kan foretages. Større LCA-databaser og datatilgængelighed er ligeledes en afgørende faktor. Det er i overensstemmelse med, hvad der nationalt arbejdes på, da softwareneutral integration mellem bygningsmodeller og LCA-beregningsværktøjer skal medvirke til, at kompleksiteten i at udarbejde LCA-beregninger nedbringes (Indenrigs- og Boligministeriet, 2021).

I byggebranchen anvendes Industry Foundation Classes (IFC) til at udveksle BIM-data mellem software applikationer. IFC er en åben international standard, hvori et omfattende dataskema og en filudvekslingsstruktur er specificeret (buildingSMART, 2017). IFC-formatet kan dermed anvendes, når data fra BIM-software skal integreres i LCA-software. Fordelen ved at eksportere bygningsmodeller til IFC er, at data fra modellerne kan benyttes til flere formål fremfor et specifikt formål. Modsat er udfordringen, at de nødvendige parametre til at opbevare data relevant for LCA-beregningen ikke er indeholdt i IFC-skemaet (Figl et al., 2019; Theißen et al., 2020; Deng & Lu, 2023). Den manglende interoperabilitet mellem bygningsmodeller og LCA-værktøjer er en stor udfordring, da det er en central del af at beregne bygningers klimapåvirkning (Soust-Verdaguer et al., 2017; Santos et al., 2019; Teng et al., 2022). På baggrund af den identificerede problemstilling stilles forskningsspørgsmålet:

- Hvordan kan IFC understøtte udarbejdelsen af LCA-beregninger og forbedre dataudvekslingen mellem bygningsmodeller og LCA-beregningsværktøjer?

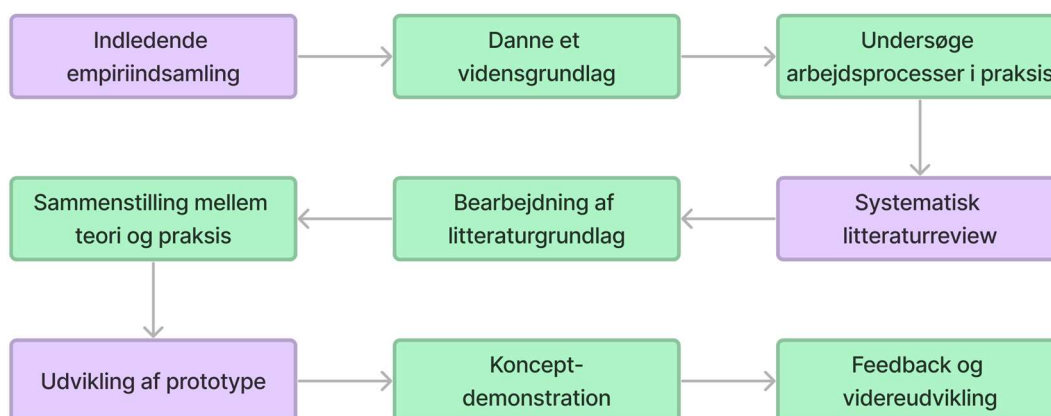
Yderligere stilles følgende underspørgsmål:

- Hvilke data er nødvendige at udtrække fra bygningsmodeller til LCA-beregninger, og hvordan fungerer den nuværende dataudveksling?
- Hvilke begrænsninger har IFC-skemaet i LCA henseende, og hvordan kan de overkommes?
- Hvordan er datastrukturen i LCA-beregningsværktøjer, og hvordan er den forskellig fra IFC?

2. Metode

I metodeafsnittet introduceres det, hvordan undersøgelsen af forsknings- og underspørgsmålene metodisk er grebet an. Først præsenteres metodens indhold og fremgangsmåde, hvorpå en mere detaljeret forklaring gives i underkapitlerne.

Specialets metodik tager udgangspunkt i grundelementerne fra Contextual Design (CD), hvor brugerinddragelse og en omfattende kontekstforståelse er byggestenene for et succesfuldt produktdesign. Ifølge forfatterne selv, Holtzblatt og Beyer (2017), er drivkraften bag CD realiseringen af, at et produkt altid vil være en del af en større sammenhæng. Værdiskabelsen sker, når der er samhørighed mellem alle produkter og processer relevant for brugens arbejdsproces. Inden for specialets tidsramme er et komplet CD-forløb efter Holtzblatt og Beyer retningslinjer ikke realiserbart. I stedet er en mere tidspassende metode udarbejdet, hvor brugerinddragelse og kontekstforståelse forbliver centrale fokuspunkter. Metoden består af tre overordnede trin: Indledende empiriindsamling, systematisk litteraturreview og udvikling af prototype. Metoden er illustreret i figur 1, hvor der for hvert trin (lilla) er udspecificeret kritiske aktiviteter (grøn).



Figur 1. Illustration af metodens workflow.

I nedenstående underafsnit uddybes fremgangsmåden og formålet for hvert trin og kritisk aktivitet.

2.1 Indledende empiriindsamling

Det første trin i metoden er den indledende empiriindsamling, der har et todelt formål:

1. At danne et vidensgrundlag og blive bekendt med fagbegreber vedrørende emnet.
2. At undersøge en brugers arbejdsprocesser i praksis

Før kontakt tages til brugeren, er det vigtigt at danne sig et tilstrækkeligt vidensgrundlag på emneområdet, så relevante spørgsmål kan stilles til brugeren. Det indebærer at blive bekendt med fagbegreber og problemstillinger på området samt forstå nødvendige arbejdsprocesser. Endvidere er mængden af litteratur på området vigtigt at kende til, inden en systematisk litteratursøgning kan foretages (Booth et al., 2016).

Når et tilstrækkeligt vidensgrundlag er opbygget, kan der tages kontakt til en udvalgt bruger. I undersøgelser af brugercentrerede udfordringer er det vigtigt at forstå, hvilke specifikke processer der skaber udfordringer hos brugeren. For at dokumentere udfordringerne i den rette kontekst bør brugerens arbejdsprocesser observeres hos brugeren, hvilket i CD kaldes Contextual Inquiry. Holtzblatt og Beyer har kort og præcist beskrevet, hvordan et brugerbesøg skal udføres: "go to the user, watch them do the activities you care about, and talk with them about what they're doing right then" (Holtzblatt & Beyer, 2017).

2.2 Systematisk litteraturreview

Det næste trin i metoden er at udføre et systematisk litteraturreview, der skal viderebygge det vidensgrundlag, som blev dannet under den indledende empirisamling. Ifølge Booth et al. (2016) kan et systematisk litteraturreview udføres af adskillige årsager, heriblandt årsagen for dette speciale, at identificere hvad der tidligere er blevet undersøgt på emneområdet, så eksisterende undersøgelser ikke duplikeres. Det er valgt, at litteraturreviewet skal være systematisk, fordi det skaber gennemsigtighed af processen for læseren, reducerer potentielle bias og simplificerer fremgangsmåden for den udførende (Booth et al., 2016).

2.2.1 Bearbejdning af litteraturgrundlag

En del af at udføre et systematisk litteraturreview omhandler at bearbejde dataet fra de studier, der identificeres igennem litteratursøgningen. I kvalitative reviews er formålet at identificere alle fællestræk fra datagrundlaget for at undersøge, om nye indsigter forekommer. Booth et al. (2016) påpeger dog, at der ikke skabes værdi ved blot at identificere og samle relateret data. Det handler om at analysere på studierne fund, metoder og anbefalinger, for at finde nye huller i litteraturen som kan undersøges. Hullerne repræsenterer den teori, som efterfølgende skal holdes op imod erfaringerne fra praksis.

2.2.2 Sammenstilling mellem teori og praksis

I sammenstillingen mellem teori og praksis skal der analyseres på, hvorvidt der i litteraturen eksisterer løsninger til brugerens udfordringer. Endvidere skal det vurderes, hvilke udfordringer der skaber mest værdi for brugeren at løse. Idégenerering af løsningsmuligheder samt overvejelser om, hvorvidt løsningsforslagene er tekniske mulige, bør ligeledes foretages i sammenstillingsprocessen. Sammenstillingen skal afsluttes med en afgrænsning af, hvad der inden for specialets tidsramme er realistisk at opnå. Afgrænsningen danner rammerne for udviklingen af prototypen, og skal derfor skarpt definere, hvilke udfordringer, i hvilken kontekst, prototypen skal løse. Holtzblatt og Beyer (2017) anbefaler at bruge grafiske repræsentationer fremfor tekst til at præsentere afgrænsningen, fordi den samlede informationsmængde bliver lettere at forstå. Informationer i tekstform læses og analyseres sekventielt, hvilket er sværere at processere, end hvis den overordnede struktur og sammenhæng kan visualiseres på samme tid.

2.3 Udvikling af prototype

Metodens sidste trin er at udvikle en prototype af et produkt, der løser brugerens udfordring. Et produkt skal ikke forstås i dets bogstavelige forstand, men derimod som en term for eksempelvis et teknisk system, en optimering eller et redesign af en funktion, en applikation eller lignende IT-ydelser (Holtzblatt & Beyer, 2017). Grundet specialets korte tidsramme er der valgt en iterativ tilgang til planlægningsprocessen, hvor identificeringen af nødvendige funktioner og elementer sker imens prototypen udvikles. Det vægtes højere at udarbejde en fungerende prototype, fremfor grundigt at planlægge udviklingen, uden en fungerende prototype. Såfremt kritiske funktioner mangler i prototypen, forventes det, at disse opdages under konceptdemonstrationen eller til feedback og videreudvikling.

2.3.1 Konceptdemonstration

Den første validering af prototypen og dens kunnen sker ved en konceptdemonstration hos brugeren, hvor det præsenteres, hvilke af brugerens udfordringer prototypen skal løse. Yderligere skal idéer og tanker om prototypens fremtidige kunnen fremlægges, så brugeren involveres i udviklingsprocessen og kan komme med inputs. Konceptdemonstrationen fungerer dermed som en kontrolproces for, hvorvidt den nuværende og fremtidige prototype er og bliver værdiskabende for brugeren. Holtzblatt og Beyer (2017) anbefaler, at brugeren sættes ind i processen, og bliver medudvikler, fremfor blot at se et produkt, da det vil skabe en mere brugbar og pålidelig feedback. De påstår, at det ellers ikke kan garanteres, at produktet løser reelle udfordringer. Inden konceptdemonstrationen finder sted, er det derfor afgørende at have udviklet en del af prototypen og samtidig have gennemtænkt slutresultatet.

2.3.2 Feedback og videreudvikling

Metodens sidste aktivitet er at præsentere den komplette prototype for brugeren. Det sker efter, at indsigterne fra konceptdemonstrationen er blevet bearbejdet og implementeret i prototypen. Hovedformålet med præsentationen er at få svar på, hvorvidt prototypen løser brugerens udfordring. Prototypen skal ikke anses som et endeligt produkt, men derimod som forarbejdet til den endelige produktudvikling. Et vigtigt element i brugercentrede udviklingsforløb er løbende at vende tilbage til brugeren for at validere udviklingen, da manglende regelmæssig kontakt med brugeren er den primære årsag til indbyrdes konflikter (Holtzblatt & Beyer, 2017). Præsentationen af prototypen betragtes derfor som en valideringsproces, hvor brugeren får mulighed for at give feedback og komme med indvendinger til videreudviklingen.

3. Analyse

I analyseafsnittet introduceres først den indledende empiriindsamling (afsnit 3.1), hvorefter valg af søgeteknik til det systematiske litteraturreview samt omfanget af litteratursøgningen beskrives i afsnit 3.2. Essentielle begreber for emneområdet, anvendt i litteraturen, forklares i afsnit 3.3, inden litteraturgrundlaget gennemgås og bearbejdes i afsnit 3.4. Analyseafsnittet afsluttes med at sammenstille dokumenterede udfordringer fra litteraturreviewet med identificerede udfordringer fra den indledende empiriindsamling (afsnit 3.5).

3.1 Indledende empiriindsamling

Den indledende empiriindsamling er opdelt i to underafsnit, hvor det første beskriver, hvad der indledningsvist er gjort for at blive bekendt med emneområdet. I det sidste afsnit præsenteres brugeren samt undersøgelsen af brugerens arbejdsprocesser i praksis.

3.1.1 Dannelse af vidensgrundlag

Ustrukturerede søgninger på Google Scholar og Scopus blev anvendt til at identificere studier som undersøgte integrationen mellem BIM og LCA, hvilke muligheder og begrænsninger integrationen har. Søgningerne frembragte adskillige litteraturreviews, der opsamlede den nyeste forskning på området, heraf et systematisk litteraturview af Teng et al. (2022) om, hvordan BIM integreres ind i LCA. Litteraturreviewet af Teng et al. blev valgt som udgangspunkt for studiets systematiske litteratursøgning (uddybes i afsnit 3.2). Udover ustrukturerede søgninger blev der eksperimenteret med LCA-beregningsværktøjet LCAByg, så en forståelse af LCA-beregningers forudgående processer kunne opnås. Endvidere blev mulighederne for at integrere data i LCAByg undersøgt, da manglende integrationsmuligheder kan være årsag til ikke at anvende beregningsværktøjet. Det viste sig, at JSON-formatet (JavaScript Object Notation) kan bruges som tredjepartsintegration i LCAByg, hvilket muliggør en effektiv dataintegration.

3.1.2 Undersøgelse af arbejdsprocesser

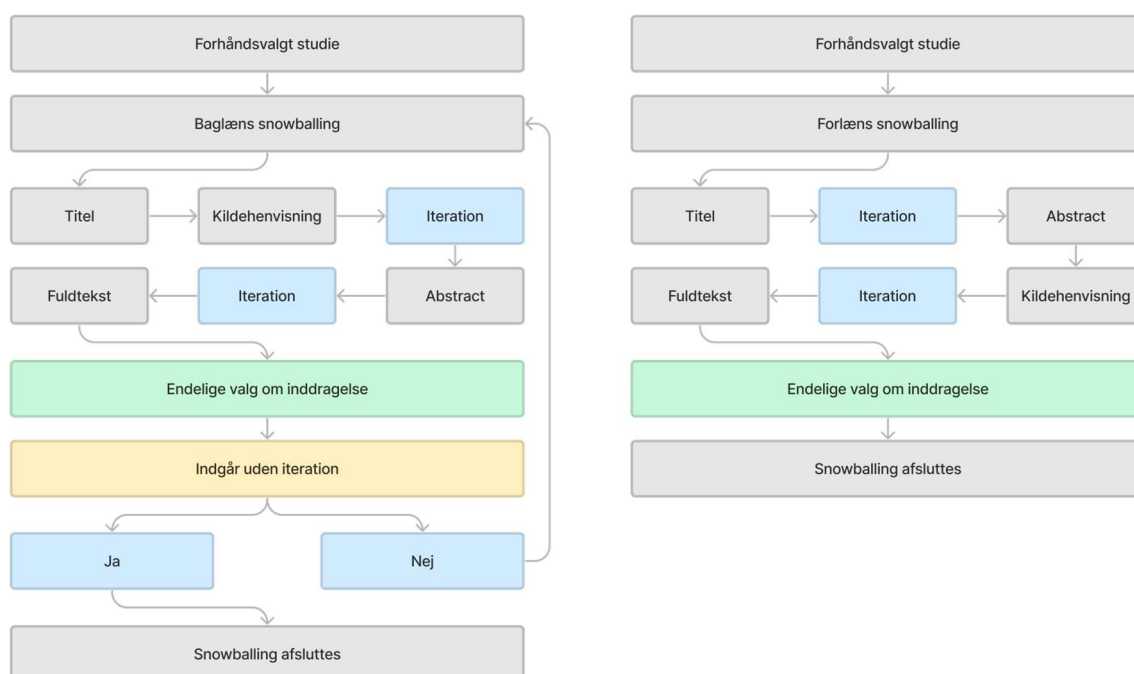
En stor rådgivende arkitekt- og ingeniørvirksomhed indvilligede i at præsentere, hvilke processer virksomheden gennemgik i udarbejdelsen af LCA-beregninger. Når der i specialet henvises til brugeren, refereres der til den rådgivende arkitekt- og ingeniørvirksomhed.

Brugerens LCA relaterede arbejdsprocesser blev, i et online møde, fremvist af en LCA-specialist, hvorefter en dybdegående dialog om processerne blev indledt. Specialisten fortalte om, og visualiserede, konkrete udfordringer, der hæmmede specialistens workflow. Ligeledes blev ønsker til et fremtidigt workflow samt potentielle løsningsforslag til udfordringer drøftet. I appendiks 1 findes en transskription af samtalen, der udspecificerer arbejdsprocesserne og de konkrete udfordringer. En opsummering af udfordringerne bliver inddraget i afsnit 3.5, hvor brugerens arbejdsprocesser sammenstilles med, hvad der står skrevet i litteraturen. Indholdt i appendikset er også gengivelser fra yderligere to møder angående brugerens LCA-proces. Disse møder blev observeret hos brugeren uden indblanding undervejs. En briefing om mødernes formål blev givet forinden, og efterfølgende var det muligt at stille opklarende spørgsmål. Møderne underbyggede mange af de samme udfordringer, som blev belyst under samtalen med LCA-specialisten.

3.2 Systematisk litteraturreview

Afsnittet indledes med at beskrive fremgangsmåden for litteratursøgningen, der er en tilpasset version af en generisk søgeteknik. Det følges op med en gennemgang af de valg, som er taget undervejs i søgeprocessen. Afslutningsvist præsenteres resultatet af den udførte litteratursøgning.

Snowballing blev valgt som søgeteknik i det systematiske litteraturreview, med en alternativ version af Wohls (2014) snowballing-proces som fremgangsmåde (se figur 2). I den alternative version bibeholdes en systematisk fremgangsmåde, men et subjektivt valg er tilføjet til snowballing-processen (udddybes senere i afsnittet). Ændringerne er tilført for at fremskynde snowballing-processen, så den tidsmæssigt er mindre omstændig end Wohls version.



Figur 2. Illustration af snowballing-processen (venstre baglæns, højre forlæns).

Under den indledende empiriindsamling blev et systematisk litteraturreview af Teng et al. (2022), om integrationen af BIM ind i LCA, identificeret. Litteraturreviewet blev i litteratursøgningen anvendt som det forhåndsvalgte studie, der skulle påbegynde baglæns snowballing-processen. Processen starter med, at alle referencer i det forhåndsvalgte studie gennemgås for at vurdere, hvorvidt studierne skal indgå i litteraturgrundlaget. Det foregår igennem iterationer for at effektivisere processen, da en grundig gennemgang af alle refererede studier vil tage alt for lang tid. Med iterationer menes der, at en aktivitet gennemføres for alle referencer inden den næste aktivitet starter. Den første aktivitet er at vurdere, hvorledes et refereret studie er relevant ud fra dens titel. Hvis det ikke kan vurderes ud fra titlen, kigges der på, hvor kildehenvisningen er

placeret i teksten på det forhåndsvalgte studie. Alle studier, som blev vurderet relevant, kan ses i appendiks 2, hvor titel, udgivelsesår og forfattere er nedskrevet. I appendikset kan det ses, hvor i snowballing-processen et studie er blevet frasorteret efter den første iteration på titel og kildehenvisning.

Studierne, der overlevede den første iteration, vurderes dernæst på deres abstract. Hvis de efterfølgende stadig er relevante, sker sidste iteration på fuld tekst. Efter studierne er læst i fuldt format, skal valget om inddragelse i litteraturgrundlaget tages. Den hidtil beskrevet baglæns snowballing-proces flugter med Wohls (2014) retningslinjer, men et subjektivt valg er hernæst tilføjet. Det skal beslutes, om studierne skal indgå med eller uden iteration. I Wohls version skal snowballing-processen genstartes for alle inddragede studier, hvilket gentages indtil ingen nye studier identificeres. I denne version kan det springes over at genstarte snowballing-processen for studierne, hvis det ønskes. Det er visualiseret i appendiks 2, hvorvidt studierne indgår med eller uden iteration. I de tilfælde hvor det blev valgt ikke at genstarte processen, var den primære årsag at studierne afveg en smule fra emneområdet, men indeholdte brugbare informationer som kunne perspektiveres til.

I forlæns snowballing anvendes studiers citeringer til at identificere ny litteratur. Fremgangsmåden er ellers meget lig baglæns snowballing, men som det kan ses i figur 2, så sker iterationerne ikke efter samme aktiviteter. Rækkefølgen af aktiviteter stemmer overens med Wohls (2014) version af forlæns snowballing, men i hans version anvendes iterationer ikke. Det blev valgt at tilføje iterationer i denne version, så et indblik i processens omfang kan visualiseres, og frasorteringer kan begrundes (appendiks 2). Endvidere er det prioriteret, som et tidsbesparende element, at udføre forlæns snowballing på nye forhåndsvalgte studier fremfor at genstarte snowballing-processen på de inddragede studier. Udvalgte identificerede studier fra forlæns snowballing er blevet anvendt som forhåndsvalgte studier til baglæns snowballing. Der har dermed været en løbende udvælgelse af forhånds valgte studier for både for- og baglæns snowballing.

Det fulde overblik af den systematiske litteratursøgning fås i appendiks 2, men den korte version gengives her. I alt blev tre omgange af baglæns snowballing og fire omgange af forlæns snowballing udført. Igennem litteratursøgningen blev 34 studier identificeret og inddraget i litteraturgrundlaget. Studierne er udgivet inden for de sidste 20 år, hvoraf størstedelen er udgivet efter år 2018.

3.3 Begrebsafklaringer

I litteraturgrundlaget findes et hav af forskellige begreber relevant for BIM-LCA integration. Fremfor at forklare begreberne undervejs gives i dette afsnit korte forklaringer på, hvad begreberne betyder. Afsnittet fungerer dermed som et opslagsværk, der kan springes over, hvis læseren er bekendt med begreberne.

3.3.1 Life cycle inventory (LCI)

Et trin i LCA der omfatter at samle centralt data for produktsystemet og kvantificere in- og outputs (DS/EN ISO 14040, 2008). En LCI kan betragtes som samlingsstedet for alle informationer relevant for LCA'en, hvilket indebærer bygningsobjekter, driftsforbruget, diverse arealer, betragtningsperioden og lignede parametre. I byggeprojekter er opdateringen af bygningsobjekter i LCI'en, deres mængder og materialer, en tidskrævende proces, da objekterne under projekteringsfaserne konstant ændrer sig.

3.3.2 Bill of Quantities (BoQ)

Et dokument hvori genstande, relevant for et specifikt formål, systematisk er samlet (Nadeem et al., 2015). I BIM-LCA sammenhæng udarbejdes en BoQ for at samle informationer fra alle bygningsobjekter i en bygningsmodel, deres mængder og evt. materialer, hvorefter informationerne eksporteres til LCI'en. BoQ kan sidestilles med et detaljeret mængdeudtræk fra bygningsmodellen, der anvendes som en del af datagrundlaget for LCA-beregningen.

3.3.3 Environmental Product Declaration (EPD)

En standardiseret proces der anvendes til at kommunikere verificerede, præcise og ikke misledende miljømæssige informationer om produkter og deres anvendelse (DS/EN ISO 15804, 2019). De miljømæssige informationer inddeles i forskellige moduler, hvor modulerne A1-3 (produktion), C1-4 (endt levetid) og D (genanvendelse) skal deklareres for alle konstruktionsrelaterede produkter. EPD-dataet fra bygningsobjekter, produktspecifikt eller generisk, bruges i LCA-beregningen til at fastslå klimapåvirkningen fra bygningsobjekterne.

3.3.4 Life-cycle costing (LCC)

En procedure til at identificere og kvantificere de betydelige omkostninger for fysiske aktiver igennem deres levetid (Woodward, 1997). Betydelige omkostninger kan eksempelvis være anskaffelse, drift, vedligehold og bortskaffelse. På byggeprojekter kan LCC anvendes som et analyseværktøj til at øge gennemsigtigheden af, hvilke byggevarer der på længere sigt er økonomisk mest fornuftige. LCC skal ansues som et helhedsorienteret værktøj til at betragte en vare totaløkonomisk.

3.3.5 Level of Development (LOD)

En international standard der fastsætter hvilke informationer fra bygningsdele i en bygningsmodel, der skal være til stede på hvilke tidspunkter under projekterings- og udførelsesprocessen (DiKon et al. 2022). I dansk kontekst er DiKon, BIM7AA og Molio blevet enige om LOD-niveauet for udvalgte bygningsdele, hvilket indebærer deres LOR-, LOG- og LOI-niveau (Level of Reliability, Geometry og Information). Ved at anvende LOD-niveauer på byggeprojekter sikres et bundniveau for, hvordan bygningsobjekters geometriske repræsentation er, hvilke informationer der er tilgængelige samt troværdigheden af de informationer.

3.3.6 Model View Definition (MVD)

En buildingSMART standard der anvendes til at definere den delmængde af IFC-skemaet, der er nødvendig for dataudvekslingen til en specifik use case eller et workflow (van Berlo, 2019). Fordi IFC-skemaet er en omfattende størrelse, anvendes MVD'er til at begrænse og specificere kravene til, hvor meget af IFC-skemaet software skal kunne håndtere. Når bygningsmodeller skal eksporteres fra BIM-software til IFC, anvendes en MVD til at definere grænsen for, hvilke informationer der bliver eksporteret.

3.3.7 Information Delivery Manual (IDM)

En buildingSMART metode der kan bruges til at dokumentere eksisterende eller nye processer samt beskrive informationen tilknyttet processerne i udvekslingen mellem forskellige parter, igennem alle faser af et byggeprojekt (DS/EN ISO 29481-1, 2017). I software sammenhæng bruges en IDM til dokumentere, hvilke informationer der skal udveksles digitalt, for at byggetekniske softwareapplikationer kan forstå og anvende informationerne. En IDM danner dermed grundlaget for, at informationsudvekslingen mellem softwareapplikationerne er pålidelig og tilstrækkelig.

3.3.8 Exchange Requirements (ERs)

En definering af informationer nødvendig for at understøtte en specifik business proces på et bestemt tidspunkt i et projekt (DS/EN ISO 29481-1, 2017). ERs er en integreret del af en IDM og bruges som basis for at udarbejde MVD'er. Formålet med ERs er at beskrive udvekslingskravene, der eksisterer, når information fra en proces af en aktør skal formidles til en anden aktør til en anden proces, i et ikke-tekniske sprog, så kravene er forståelige for aktørerne.

3.3.9 Application Programming Interface (API)

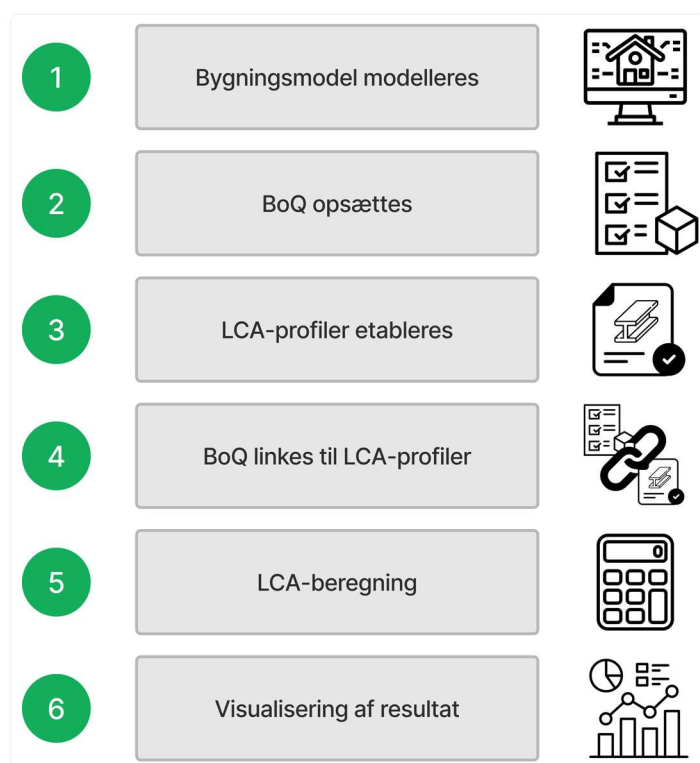
En programmeringsgrænseflade der muliggør en person eller et system at interagere med hard- eller software igennem tilgængelige operationer (Dominte, 2023). Adskillige typer af API'er findes, men i BIM-LCA sammenhæng er Web API den centrale. Et Web API indeholder nogle funktioner, der tillader en bruger at sende forespørgsler ud til en webserver, f.eks. om at lede efter noget bestemt data i en database. Hvis dataet i en database er tilgængeligt, vil brugeren kunne indhente dataet og anvende det. Brugen af et API giver dermed brugeren adgang til potentielt store mængder data, uden at skulle opbevare dataet på egne maskiner.

3.4 Bearbejdning af litteraturgrundlag

Studierne, der udgør litteraturgrundlaget, berører adskillige emneområder, der på hver sin måde bidrager til en bedre forståelse af, hvad der tidligere er undersøgt eller udviklet i samspillet mellem BIM og LCA. Studierne er blevet kategoriseret efter følgende emneområder: Integrationsstrategier, Perspektivering til energiområdet, Tidligere udarbejdede BIM-LCA-værktøjer og Anbefalinger til et fremtidigt workflow. I dette afsnit præsenteres ny viden, anbefalinger, bedrifter, udfordringer og lignende fra studierne.

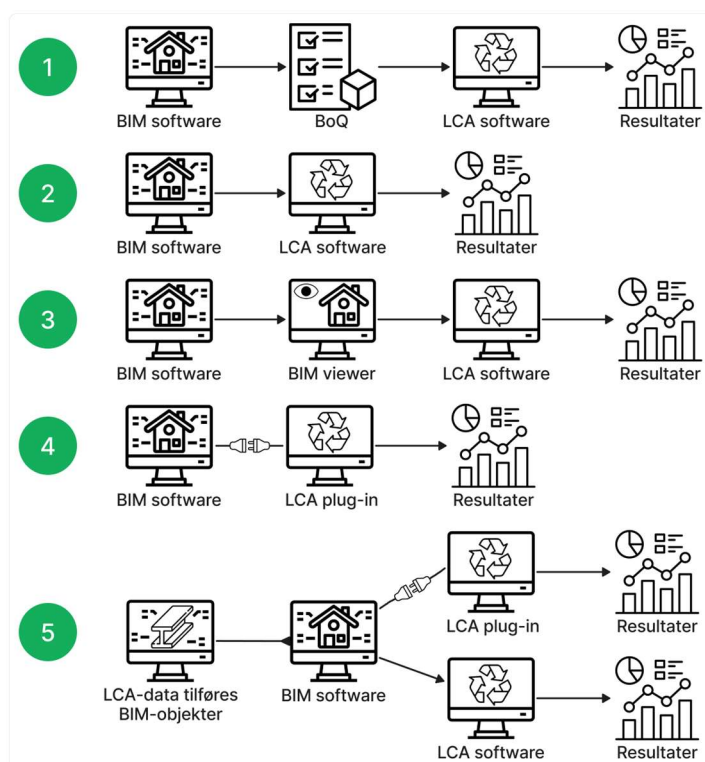
3.4.1 Integrationsstrategier

I gennem årene er adskillige strategier blevet afprøvet, i forsøget på at ramme det mest optimale workflow for at integrere BIM-software med LCA-software. I 2019 sammenfattede Wastiels og Decuyperes de forskellige BIM-LCA integrationsstrategier, og definerede de nødvendige trin for at anvende BIM-informationer til at udføre en LCA-beregning. De identificerede seks nødvendige trin, der er illustreret i figur 3. I det første trin modelleres bygningsmodellen af bygningsobjekter med tilknyttet egenskabsdata (Building Information Model). Herefter opsættes en Bill of Materials (BoM), der etableres ud fra bygningsmodelles elementer og/eller materialer. Trin tre er at etablere LCA-profiler, der dokumenterer de miljømæssige påvirkninger, hvilket kan ske på forskellige niveauer såsom materiale-, komponent- eller bygningsdelsniveau. Der kan ligeledes anvendes generisk data eller produktspecifikke miljøvaredeklarationer (EPD'er). Med BoM og LCA-profilerne på plads skal de efterfølgende linkes sammen, så påvirkningen fra hvert element kan defineres. Det er ikke nødvendigvis et trin for sig selv, da det kan gøres samtidig med, at LCA-profilerne bliver etableret. Til slut udføres en LCA-beregning, efter reglerne foreskrevet i LCA-værktøjets beregningskerne, og resultatet visualiseres, så den LCA-ansvarlige kan analysere og videreformidle slutresultatet.



Figur 3. Wastiels og Decuyperes (2019) seks nødvendige trin for at bruge BIM-informationer til en LCA-beregning.

Af BIM-LCA integrationsstrategier definerer de fem forskellige, der i år 2023 stadig bliver refereret tilbage til (Guignone et al. 2023, Tam et al. 2023, Forth et al. 2023). De fem strategier skiller sig ud fra hinanden ved at have et forskelligt workflow, hvilket kan ses i figur 4.



Figur 4. Wastiels og Decuypere (2019) fem integrationsstrategier i en forsimplet version.

- 1) BoQ udtrækkes fra BIM-software til et regneark (ofte), hvorefter det importeres ind i LCA-software og linkes manuelt sammen med LCA-profilerne. LCA-beregningen og visualisering af resultaterne sker i LCA-softwaremiljøet.
- 2) Bygningsobjekters geometriske parametre samt relevant egenskabsdata importeres ind i LCA-software ved brug af IFC (ofte). I LCA-softwaremiljøet linkes LCA-profilerne til bygningsobjekterne, efterfulgt af beregning og resultatvisning.
- 3) Samme fremgangsmåde som strategi to, men med et mellemliggende step, hvor LCA-profilerne linkes til bygningsobjekterne i en BIM-viewer, der indeholder samme LCA-profiler som dem i LCA-softwaren.
- 4) Et LCA plug-in anvendes, hvor LCA-profiler kan linkes til bygningsobjekterne i BIM-softwaremiljøet. LCA-beregningen og visualisering af resultaterne sker i LCA plug-in'et.
- 5) Bygningsobjekter beriges med LCA-data i BIM-softwaremiljøet, som LCA plug-in'et eller LCA-softwaren kan genkende og sammenkoble med LCA-profiler. Beregning og resultatvisning sker i LCA-softwaremiljøet eller i plug-in'et.

Der kan findes fordele og ulemper ved alle strategierne, men i og med at dette speciale har IFC som et omdrejningspunkt, er strategi nummer to, tre og fem mest relevant.

3.4.2 Perspektivering til energiområdet

I bygningsreglementet blev krav til bygningers energiforbrug indført før klimapåvirkningskrav. Det undersøges derfor, hvad der igennem årene på energifronten er forsøgt udviklet og implementeret for at fremme udarbejdelsen af bygningers energirammer. Der perspektiveres ikke til energispecifikke udfordringer, men derimod til dataudvekslings- og datahåndteringsudfordringer.

Et studie af Ma og Zhao (2008) undersøgte, hvordan energibesparelser kunne opnås ved at designe energieffektivt. De præsenterede et workflow, som de mente var ideelt for den næste generations energieffektivt bygningsdesign-software. I workflowet blev IFC brugt som dataudvekslingsformat mellem forskellige applikationer, der skulle forbindes igennem en håndteringsplatform med alt BIM-data. Formålet med at benytte en håndteringsplatform var, at data kunne tilgås på en ensartet måde, fremfor at der skulle udvikles applikationsspecifikke løsninger. Ma og Zhao påpegede, at ens datastruktur mellem BIM-data og IFC var nødvendigt, og at IFC-skemaet måtte udvides, hvis BIM-egenskaber manglede i det. Pinheiro et al. (2018) noterede sig, at data fra BIM skulle genindtastes manuelt i energiberegningssoftware, fordi data gik tabt ved IFC-eksporten af BIM. Endvidere manglede vigtige komponenttyper for energiberegningen, såsom varmepumper, at blive defineret i IFC4-skemaet, da standardiseringsprocessen er kompleks og tidskrævende. Pinheiro et al. demonstrerede derfor, hvordan IDM og MVD kunne anvendes som en standardiseret metode for dataudveksling på energiområdet. De identificerede manglende IFC-egenskaber og komponenttyper ved at gennemgå adskillige projekter med varierende kompleksitet, hvorefter egenskaberne og typerne blev tilføjet til deres MVD igennem IfcDoc-værktøjet. Værktøjet tillader brugeren at definere regler for specifikke typer (entities) og deres attributter, så det i et udvekslingsscenarie kan sikres at datainputtet er validt. Ligeledes kan værktøjet generere HTML-dokumentationen nødvendig for, at funktioner i MVD'er kan importeres og eksporteres af BIM-applikationer. Pinheiro et al. betragtede deres IFC-baserede dataudvekslingsmetode som en succes og konkluderede, at deres arbejde kunne anvendes som et udgangspunkt for videreudviklingen af en officiel buildingSMART MVD.

Andriamamonjy et al. (2018) præsenterede en meget lignende IFC-baseret dataudvekslingsmetode, men i stedet for at benytte IfcDoc-værktøjet udviklede de et Python-baseret værktøj (Ifc2Modelica). Deres studie efterviste, ligesom Pinheiro et al., at IFC-standardens fleksibilitet nok til at blive anvendt i energisimuleringer, fordi brugerdefinerede egenskabsæt, regler og MVD'er kan udformes. I et studie af Xu et al. (2019) blev IFC-skemaet også udvidet igennem brugerdefinerede egenskabsæt, så data om energiforbruget, i fremstillingen af præfabrikerede bygningsdele, kunne mappes til en IFC-model. Python-programmering blev anvendt til at udarbejde et analyseværktøj, der kunne udtrække energidataet fra IFC-modellen og herefter udføre en energiforbrugsanalyse. Formålet med analyseværktøjet var at visualisere, hvordan energiforbruget fordelte sig

imellem produktion af råmateriale, transport af råmateriale og fabriksproduktion af præfabrikeret bygningsdel. En visning af forskellige energiforbrugsscenarier giver projekterende en større indsigt i deres muligheder for valg af bygningsdele og materialer.

Et optimeringsforslag til dataudveksling blev introduceret af Lai et al. (2019), der fokuserede på, hvordan dataudvekslingen kunne forbedres i tilfælde, hvor ingen relevant IDM/MVD eksisterede. De foreslog at anvende en ER (Exchange Requirements) matrix til at effektivisere defineringen af ERs, ved at definere flere ERs på en gang fremfor enkeltvis. Matrixen skulle herefter bruges til automatisk at mappe ERs til IFC-data, så en IFC-udvekslingsmodel kunne genereres ved at bruge en algoritme. De begrænsede brugen af deres algoritme til konstruktionsmæssigt design, men konkluderede at den kunne udbredes til andre domæner.

3.4.3 BIM-LCA værktøjer

I dette afsnit beskrives adskillige BIM-LCA værktøjer, der igennem tidens løb er udviklede til at optimere de eksisterende arbejdsprocesser iht. LCA i byggebranchen. Værktøjerne bliver gennemgået kronologisk efter udgivelsesår for at belyse, hvordan de har udviklede sig.

Det ældste BIM-LCA værktøj fra litteraturgrundlaget er LCADesign, der blev udviklet af Tucker et al. (2003). Værktøjet var i stand til at oprette LCA-databaser med specifikke bygningsvarer, der kunne mappes sammen med bygningsobjekter igennem et EXPRESS Data Manager (EDM) system. Dets formål var at fungere som et hjælpeværktøj til at tage bedre miljø- og prismæssige beslutninger i projekteringen af kommercielle bygninger. Bygningsobjekterne blev, igennem en IFC-eksport af bygningsmodellen fra en CAD-applikation (Computer-Aided Design), indført i EDM-systemet, hvorfra alt blev styret (beregning, resultatvisning, mapping osv.). Til trods for mange manuelle processer samt at IFC var relativt nyt i år 2003, var det muligt at udvikle et brugbart BIM-LCA værktøj.

Loh et al. (2007) så ligeledes potentialet i at anvende IFC som en fælles standard for bygningsmodeller og vurderinger af bygningers klimapåvirkning. Fremfor at indtaste data i både en bygningsmodel og i et miljøvurderingsværktøj kunne det gøres i én omgang, ved at tilføje egenskabsdata til bygningsdele. Loh et al. introducerede derfor et konceptuelt rammeværktøj, der tilføjede miljødata, fra en central database, til bygningsdele som egenskabsdata. Et miljøvurderingsværktøj skulle herefter indhente og anvende miljødataet fra de berigede bygningsdele til en miljøpåvirkningsberegning. Resultaterne fra beregningen skulle til slut linkes tilbage til bygningsmodellen igennem en IFC-viewer. Rammeværktøjet skulle medvirke til hurtigere at kunne beregne miljøpåvirkningerne, da enhver lille ændring i bygningsdesignet kræver en ny beregning, hvilket er en tidskrævende opgave.

Samme tidsoptimering søgte Jrade og Abdulla (2012), da de brugte IFC som dataudvekslingsformat til at integrere LCA med BIM. Deres workflow bestod af at berige bygningsdele i Revit med et miljøtag tilsvarende en bestemt konstruktionsopbygning, udspecificeret i en LCA-beregner. Bygningsdelene blev dernæst eksporteret i IFC-formatet fra Revit til Excel, hvor en makro var udviklet til at søge efter miljøtagget i bygningsdelenes egenskabsdata. Hvis bygningsdelene havde et miljøtag, blev deres arealer og volumener udtrykt og konverteret til kvadratmeter. Kvadratmeterne blev så overflyttet til et regneark, hvorfra LCA-beregningsen fandt sted, og resultaterne blev visualiseret. Workflowet er et eksempel på integrationsstrategi nummer to, hvor bygningsobjekters geometriske parametre bliver importeret ind i LCA-software. I stedet for at linke LCA-profiler til bygningsobjekterne i LCA-softwaremiljøet blev de linket på forhånd i Revit med miljøtagget. Jrade og Abdulla nævner afslutningsvis i deres artikel, at en fremtidig videreudvikling kunne være at integrere API'er i BIM-softwaremiljøer, så resultater fra LCA-beregninger kan blive mere interaktive og brugervenlige.

Kulahcioglu et al. (2012) lykkedes med at gøre deres LCA-resultater interaktive, da deres værktøj var i stand til at sammenligne forskellige scenarier, hvorved effekten af ændringer i bygningsdesignet nemmere kunne visualiseres. De brugte Blender som deres udviklingsplatform, hvortil data fra en IFC-model blev importeret igennem Blenders Python API. Fra Blender blev IFC-modellens bygningsobjekter og tilhørende egenskabsdata udtrykt til et LCA-softwareprogram, der kunne udføre LCA-beregningsen. Resultatet fra LCA-beregningsen fik brugeren vist igennem det Blender-baseret interaktive værktøj, som Kulahcioglu et al. kategoriserede som et hjælpeværktøj til at tage miljørelevante designbeslutninger. En lignende fremgangsmåde anvendte İlhan og Yaman (2016) i deres GBAT-værktøj (Green building assessment tool), men fremfor at bruge Blender udviklede de deres egen GBAT-applikation. En IFC-baseret template skulle udfyldes i BIM-miljøet, der sammen med IFC-filen for bygningsmodellen blev importeret ind i GBAT-værktøjet. I værktøjet blev et program brugt til at udtrække de relevante informationer til regneark i Excel, der blev brugt som grundlag for miljøpåvirkningsberegningen.

En anden slags IFC-template blev brugt af Santos et al. (2019), da de undersøgte, hvordan informationer skal struktureres i BIM, for at fremme interoperabiliteten mellem forskellige værktøjer til analyser som LCA og LCC (Life-cycle costing). De anvendte en IDM/MVD metodik til at identificere, hvilke informationer IFC4-skemaet manglende at integrere i objekter, før en komplet LCA/LCC analyse kunne udføres automatisk. De konkluderede, at 137 IFC-egenskaber var nødvendigt, hvis deres fremgangsmåde blev anvendt. I IFC4-skemaet var blot 16 af de 137 IFC-egenskaber indeholdt, hvormed 121 IFC-egenskaber manglende at blive integreret, før en komplet LCA/LCC analyse kunne udføres ved brugen af IFC i et BIM-softwaremiljø. Santos et al. mener, at deres undersøgelse har givet et solidt fundament til fremtidige værktøjer ved at præsentere, hvilke informationer der skal udveksles mellem aktører. Softwareudviklere vil kunne lave en MVD, der kan bruges, når

informationer vedrørende LCA og LCC skal udveksles mellem software, hvilket ligeså vil fremme brugen af openBIM. Et studie af Figl et al. (2019) konkluderede ligeledes, at udfordringen med openBIM var de manglende egenskaber i objekterne. De noterede yderligere, at udfordringen var størst i HVAC-sammenhæng (Heat, Ventilation, Aircondition, Control), og at egenskaberne sjældent blev brugt i praksis.

Exchange Requirements (ERs) bruges, ifølge ISO 29481 (IDM-standard), når nødvendige informationsudvekslingskrav for en specifik proces skal identificeres. Theißen et al. (2020) definerede 222 ERs for at kunne udføre en WBLCA (Whole Building Life Cycle Assessment) på en bygningsmodel, indeholdt 7312 objekter, i et casestudie. De definerede ERs blev tilknyttet bygningsmodellens IFC-datamodel, men 13 ERs kunne ikke tilknyttes standard IFC-egenskaber. En udvidelse af IFC-skemaet (2x3) var derfor nødvendig, hvilket blev gjort manuelt ved at tilføje egenskaber til bygningsmodellen i BIM-softwaremiljøet som projektparametre. Alle objekter i bygningsmodellen fik ligeledes tilføjet et parameter kaldet "UUID" (Universally Unique Identifier), der blev brugt til manuelt at tilknytte LCA-data fra den tyske ÖKOBAUDAT database til objekterne. Grundet de tilføjede projektparametre kunne en standard MVD til IFC-eksport fra BIM-softwaremiljøet ikke udføres. En integreret eksportfunktion i BIM-softwaremiljøet blev derfor anvendt, så de tilknyttede LCA-data fra parametrene var indeholdt i IFC-datamodellen. I et LCA-softwaremiljø kunne LCA-beregningen foretages automatisk, fordi de tidligere indtastede LCA-data kunne genkendes i LCA-softwaremiljøet. Resultaterne fra beregningen kunne herefter eksporteres som en IFC- eller PDF-fil. Dette workflow minder om det tidligere præsenteret workflow fra Jade og Abdulla (2012), men veldefinerede informationsudvekslingskrav og et forbedret LCA-softwaremiljø skaber mere brugervenlighed og gennemsigtighed af workflowet. Der er dog stadig en del manuelt arbejde i at skabe en forbindelse mellem objekterne i IFC-modellen og LCA-data fra en database.

Xu et al. (2021) udarbejdede i programmeringssproget Python et dataoverførelsesværktøj som, ved brugen af IFC-skemaet, automatisk skulle etablere en forbindelse mellem objekter i et BIM-softwaremiljø og LCA-data fra en database. Værktøjet skulle reducere mængden af manuelt arbejde, der normalt var forbundet med BIM-LCA-mapping, ved at oversætte datastrukturen fra BIM til datastrukturen i et bestemt LCA-softwareprogram. Ifølge Xu et al. var deres BIM-LCA metode en signifikant forbedring til den traditionelle LCA-proces, men de anerkender samtidig, at det har nogle begrænsninger i og med, at det er udviklet til ét bestemt LCA-softwareprogram. Derudover var manuel mapping stadig nødvendigt på materialeniveau, og ændringer i BIM-softwaremiljøet blev kun opdateret i LCA-modellen ved at genimportere CSV-filerne fra dataoverførelsesværktøjet og dermed overskrive det eksisterende grundlag. I en senere udgivet artikelversion (år 2022) nævner Xu et al., at idéen med datastrukturadaption kan udvides til andre LCA-softwareprogrammer

ved at anvende forskellige templates. Ydermere at fremtidig forskning burde undersøge muligheden for to-vejs kommunikation, så data fra LCA-softwaremiljøet kan sendes tilbage til BIM-softwaremiljøet.

Idéen, fra Xu et al., om at forbedre dataudvekslingen mellem BIM- og LCA-software ved at anvende templates brugte Alwan og Jones (2022), i deres studie om en IFC-baseret fremgangsmåde for en tidlig LCA-screening. Fremfor at benytte integrationsstrategi nummer to, hvor LCA-profiler linkes til bygningsobjekter i et LCA-softwaremiljø, valgte de at berige bygningsobjekter med LCA-data i et BIM-softwaremiljø (integrationsstrategi nummer 5). LCA-dataet, om bygningsmaterialer fra en LCA-database, blev importeret ind i BIM-softwaret, således at et LCA-materialebibliotek kunne oprettes og gemmes som en template. Bygningsobjekter i BIM-softwaremiljøet skulle oprettes med materialer fra LCA-materialebiblioteket, så LCA-dataet kunne eksporteres som IFC-egenskaber. Et Python-kodet beregningsværktøj, pycab, blev udviklet til at udføre den tidlige LCA-screening, der brugte IFC-eksporten af bygningsmodellen som datainput. Beregningsværktøjet anvendte open source biblioteket IfcOpenShell, der kan bruges til at læse, skrive og modificere BIM igennem IFC, og adskillige udvidelsespakker til Python (matplotlib, numpy og pandas). Udover at udføre en tidlig og simplificeret LCA-screening kunne beregningsværktøjet præsentere alternative materialevalg fra LCA-materialebiblioteket, der ville minimere bygningens klimapåvirkning. En komplet LCA-beregning vil dog ikke kunne udføres med værktøjet, da beregningskernen er meget simplificeret. Eksempelvis anvendes blot en miljøfaktor fremfor korrekt miljødata ved bestemte objekter (vinduer, døre og trapper).

Et andet eksempel på integrationsstrategi nummer fem fremvises af LLatas et al. (2022), hvis fokus ligeså var på de tidlige stadier i designprocessen. De undersøgte, hvordan resultater fra en LCSA (Life Cycle Sustainability Assessment), der indeholder LCA, LCC og S-LCA (Social-LCA), kunne visualiseres i realtid. I den visuelle programmeringsplatform Dynamo blev IFC-egenskaber tilknyttet bygningsobjekterne fra BIM, hvorefter de blev linket til en database (Excel-fil der sammenholdt LCA-, LCC- og S-LCA-data). Efterfølgende blev bygningsobjekternes volumen udtrukket fra BIM og brugt til at udføre LCSA-beregningen, der igennem diagrammer blev udtrykt. LLatas et al. fandt frem til, at to IFC-egenskabssæt samt 33 IFC-egenskaber skulle tilføjes til IFC4-skeamet, for at Dynamo-scriptet automatisk kunne udføre en LCSA. Studiet demonstrerede at IFC-egenskabs-sættet IfcPropertySet, kan bruges til at berige IFC-formatet med brugerdefinerede informationer, hvilket i et lukket BIM-softwaremiljø ikke skaber udfordringer. Udfordringerne opstår først, når flere softwareapplikationer skal udveksle informationer imellem hinanden. Hertil kan IDM og MVD'er udarbejdes, så det sikres, at applikationerne snakker det samme sprog.

Serrano-Baena et al. (2023) anvendte en cloud-baseret platform til at opbevare data fra en bygningsmodel. Bygningsmodellen blev eksporteret til IFC igennem en officiel buildingSMART MVD, hvor alle informationerne

kunne opbevares i IFC-parametre. I studiet blev der fokuseret på materialevalg tidligt i byggeprocessen, hvorved det ikke var nødvendigt at udvide IFC-skemaet eller udarbejde en IDM. Zhang et al. (2023) integrerede også en cloud-baseret platform i deres LCA-prototypesystem, hvorfra BIM-data kunne tilføjes, redigeres, opdateres og valideres. De valgte at udvide IFC-skemaet (IFC4x3_RC3) ved at bruge ifcOWL, der kan repræsentere IFC-data i webteknologier (f.eks. Semantic Web og Linked Data), hvor dataet er tilgængeligt som RDF (Resource Description Framework) (Pauwels & Terkaj, 2016). Ræsonnementsregler og SPARQL queries blev herefter brugt til at understøtte interoperabiliteten mellem IFC-dataet og deres LCA-prototypesystem. Zhang et al. påpeger, at BIM mangler semantisk komplethed, fordi IFC-skemaet er nødsaget til at blive udvidet. Eksempelvis nævnes, at generiske modeller bruges i Revit, når korrekte objektklassificeringer ikke er tilgængelige i IFC-skemaet. Ydermere, at brugerdefinerede parametre i BIM-softwaremiljøer er svære at generalisere i standardiserede metoder.

Deng og Lu (2023) udviklede et værktøj, der kunne beregne klimapåvirkningen på forskellige niveauer (Bygnings-, domain-, element-, BoQ- og ressourceniveau) med datainput fra flere IFC-modeller. De udarbejdede en algoritme, der kunne fratrække den overlappende overflademængde mellem objekter, når objekterne fra forskellige IFC-modeller kolliderede med hinanden. Yderligere foreslog de, at informationer om klimapåvirkning i IFC-filer kunne anvendes som datatræningssæt for kunstig intelligens, hvilket de så en relevant fremtid for i tidlige klimapåvirkningsscreeninger.

Idéen om at anvende kunstig intelligens i de tidlige designstadier blev undersøgt af Forth et al. (2023). De udarbejdede et værktøj, der brugte Natural Language Processing (NLP) til at mappe IFC-objekter med LCA-data, så et fuldautomatisk workflow kunne opnås. Værktøjet skulle understøtte beslutninger om designvalg i de tidlige projekteringsstadier. Kort fortalt drejer NLP sig om, at transformere menneskeskabt tekst om til noget maskinlæsbart ved at bruge diverse teknikker. Hvert ord bliver givet en numeriskværdi, der beskriver den semantiske lighed mellem ordene. Jo mere ordene minder om hinanden, desto mindre afstand er der imellem dem. Den semantiske lighed imellem ord kan lade sig gøre, fordi ordene bliver konverteret til deres ordbogsbeskrivelse (Forth et al., 2023). En 86,72 procent gennemsnitlig succesrate over fem casestudier, på automatisk at mappe IFC-objekter med LCA-data (fra ÖKOBAUDAT), blev opnået. Succesraten var mest afhængig af den semantiske modelkvalitet, primært vedrørende objektklassifikation, hvorvidt den var valid eller korrekt. De største begrænsninger var det manuelle workflow til objektklassifikation samt lange procestider i værktøjet.

3.4.4 anbefalinger til et fremtidigt workflow

Flere litteraturreviews af BIM-LCA integrationen blev inkluderet i litteraturgrundlaget, da de sammenfatter erfaringer fra tidligere studier til at belyse tendenser og drage konklusioner. Mange anbefalinger til fremtidige udviklinger på BIM-LCA området findes derfor i litteraturreviews, og de anbefalinger præsenteres i dette afsnit. Udover litteraturreviews blev også relevante studier om dataudvekslingsoptimering inkluderet i litteraturgrundlaget, og anbefalingerne derfra bliver ligeledes præsenteret i dette afsnit.

Tam et al. (2023) valgte at tage udgangspunkt i Wastiels og Decuyperes (2019) integrationsstrategier, da de i deres studie forsøgte sig med at kvantificere, hvilken af integrationsstrategierne var mest fordelagtig i forskellige scenarier. De valgte fem parametre, som integrationsstrategierne skulle bedømmes på: data kvalitet, tidsforbrug, automatiseringsgrad, kompleksitet og niveauet af nødvendige færdigheder. De konkluderede, at strategi nummer fire (LCA-plugin til BIM) var mest fordelagtig i alle scenarier (designfasen, projekteringsfasen og udførelsesfasen), primært fordi det muliggør for at bruge visuel programmering til at automatisere processer. Hvorvidt deres undersøgelse kan bruges som argumentation for valg af integrationsstrategi kan diskuteres, da parametrene var udvalgt på baggrund af ni interviewpersoners mening. Studiet er dog et bevis på, at det holistiske billede mangler at blive vurderet, da eksempelvis ens BIM-software blandt de involveret faggrupper ikke er en selvfølge. Yderligere mangler et fokus på at inddrage LCA i den eksisterende arbejdsproces, således at informationsgrundlaget for LCA-beregningen kan sikres at være opdateret. Muligheden for at flytte BIM-data vil ifølge Di Bari et al. (2022) forbedre samarbejdet mellem forskellige faggrupper. Di Bari et al. analyserede data fra deres spørgeskema, om LCA-værktøjer og deres funktioner, for at belyse, hvad eksisterende LCA-værktøjer var i stand til, samt hvad den næste generation skulle kunne. De kom frem til, at næste generation af LCA-værktøjer skal understøtte tidlige LCA-beslutninger bedre, og samtidig kommunikere usikkerheder ved LCA-data med større gennemsigtighed. Feedback i realtid og workflows med mere automation anses ligeledes som vigtige implementeringer i værktøjerne, f.eks. plug-in eller IFC-objektberigelse.

En sammenligning af integrationsstrategiers LCA-beregningspræcision blev præsenteret af Zheng et al. (2023), hvor den IFC-baserede tilgang viste sig at være mere retvisende end plug-in tilgangen (afveg med henholdsvis +1,33% og -0,82%). Upræcise BoQ blev identificeret med plug-in tilgangen, hvorimod den IFC-baserede tilgang døjede med, at flere elementer blev kombineret til ét element i IFC-format. Zheng et al. undersøgte også tidsforbruget på at oprette en LCI ved de forskellige strategier, hvor den IFC-baserede tilgang klart tog længst tid. Sammenkoblingen mellem BoQ og LCA-profiler var den største tidsrøver, men når BoQ opdateres kan meget tid spares ved allerede at have etableret en forbindelse. Theißen et al. (2020)

foreslog at bruge en regelbaseret sammenkobling, når et link mellem LCA-data fra en database og bygningsobjekter i en IFC-model skulle matches. De integrerede UUID (Universally Unique Identifier) i både LCA-databasen og bygningsobjekterne, således at et link kunne etableres, såfremt ens UUID var tilfældet. Til trods for at deres metode reducerer tidsforbruget i at linke objekter og LCA-data, så tilføjer metoden blot tid i oprettelsen af bygningsobjekterne, da objekternes UUID blev indtastet manuelt. Studiet er dog et eksempel på de mange automatiseringsmuligheder, der opstår, når LCA-databasers API benyttes.

Klumbyte et al. (2023) tog det skridtet videre, da de i deres studie skabte et BIM til LCA API (6D BIM API). Deres workflow bestod af tre stadier: dataforberedelse, dataudtræk og 6D BIM generering. I dataforberedelse blev en BIM eksporteret til en IFC-fil, og LCA-indikatorer fra EU rammeværktøjet Level(s) blev udvalgt. Level(s) sikrer et fælles sprog til at vurdere og rapportere bygningers bæredygtighedsperformance. Level(s) er udarbejdet for at mindske forskelligheder, i måden bæredygtighed i byggebranchen bliver defineret i EU (European Union, 2021). I dataudtræk-stadiet blev Python-kodning sammen med IfcOpenShell brugt til at udtrække data fra EPD'er i LCA-databaser samt IFC-filen, hvorefter dataet blev sendt til en LCA-beregner. Til slut blev en 6D BIM genereret, hvori LCA-resultaterne var. Det innovative ved studiet ligger i Python-kodningen, da det var muligt at udtrække data fra adskillige filtyper (IFC, Excel og PDF). Klumbyte et al. påpeger, at manglende lokationsdata er en begrænsning. De mener, at lokationsdata kan forbedre brugbarheden ved at BIM-LCA værktøjer, fordi mere kontekstbevidste beslutninger kan tages.

I et studie af De Wolf et al. (2023) blev en liste over EU-tilgængeligt LCA-software og databaser oprettet og analyseret for at vise, hvad der eksisterer på markedet, samt hvad det er i stand til. Herudover beskrives det, hvorvidt softwaret eller databaserne er i overensstemmelse med EU's rammeværktøj Level(s). Den danske LCA-software, LCAbyg, indgår ikke i listen, men EPD Danmark databasen er repræsenteret. Databasen er i overensstemmelse med standarden EN 15804, men har ikke mulighed for at blive tilpasset Level(s) (part 1 og 2, part 3). Ydermere er import til og eksport fra databasen ikke muligt, dog har et samarbejde mellem EPD Danmark og BUILD (skaberne af LCAbyg) gjort det muligt at importere EPD'er fra EPD Danmark ind i LCAbyg. Studiet af De Wolf et al. bevidner, hvorfor et rammeværktøj som Level(s) er relevant, fordi listen med software og databaser er utrolig lang. Det bliver hurtigt uoverskueligt at identificere funktionalitetsforskelligheder og sammenligne resultater, hvilket er kritisk for anvendeligheden af en LCA-beregning. Valg af benchmarking system kan ligeledes have en stor indflydelse, hvilket blev eftervist af Rasmussen et al. (2022). I deres studie blev 17 forskellige cases fra 14 forskellige lande analyseret. Ligheder og forskelle i benchmarking systemerne blev analyseret, og anbefalinger til fremtidige benchmarking systemer blev præsenteret. Stor variation i klimapåvirkning blev identificeret, hvorefter det delvist blev begrundet med forskelligheder i benchmarking systemerne. Baggrundsdata og valg af værktøjer blev ligeså vurderet til at have en stor indvirkning

på klimapåvirkningen. Rasmussen et al. anbefaler derfor, at der er komplet gennemsigtighed i beregningsreglerne, og måden indholdet (BoQ, LCA-data og lignende elementer relevant for LCA-beregningen) opgøres på.

Zimmermann et al. (2021) udførte otte semistrukturerede interviews med én eller flere informanter fra forskellige ingeniør- og arkitektvirksomheder, repræsenteret i den danske byggebranche, der alle udførte LCA på bygninger. Informanterne skulle blandt andet tage stilling til, hvordan de arbejdede med BIM i relation til LCA, hvilke udfordringer de stødte på iht. BIM-LCA samt hvad de anså som vigtigst for et godt BIM-LCA workflow. Desuden havde Zimmermann et al. udarbejdet en prototype ud fra integrationsstrategi nummer tre, hvor en BIM-viewer bruges til at linke bygningsobjekter og LCA-profiler, som informanterne skulle bedømme. I prototypen blev mapping mellem bygningsobjekter og LCA-profiler gjort enten manuelt eller semiautomatisk. Til den semiautomatiske mapping kunne prototypen foreslå matches baseret på tidligere valg eller navnene på materialerne. Til LCA-beregningen blev LCAbyg-værktøjet brugt, hvor en direkte forbindelse mellem LCAbyg og prototypen blev etableret i Python. JSON-formatet blev brugt til at udveksle data med LCAbyg. Endvidere kunne prædefinerede bygningsdele (LCA-profiler) vælges til den tidlige LCA, eller specifikke materialer kunne importeres fra BIM eller oprettes i prototypen til en mere detaljeret LCA. Prototypens BIM-viewer fungerede som brugerinterface, hvorfra alle data kunne kontrolleres og visualiseres. Informanterne fortalte, at kvaliteten af bygningsmodellen ofte var et problem, fordi supplerende informationer skulle indhentes for at færdiggøre eller detaljere BoQ. En forventningsafstemning iht. korrektheden af BoQ i forskellige LCA-stadier bør derfor overvejes, så det vides, hvilke informationer skal være tilgængelige på hvilke tidspunkter. Informanterne var overvejende positive mod prototypens BIM-viewer, da de mente, at den øgede gennemsigtigheden af data samt visualiseringen af resultaterne. Det blev dog påpeget af flere informanter, at en BIM-viewer ikke ønskes, hvis den sænker mapping-processen. Til slut konkluderer Zimmermann et al., at virksomheder vil have forskellige ressourcer og målsætninger, hvormed forskellige behov i et BIM-LCA workflow må forventes.

Tre litteraturreviews om at integrere BIM med LCA blev inkluderet i litteraturgrundlaget. Soust-Verdaguer et al. (2017) bemærkede, at LOD-niveauet i bygningsmodellerne fra de inkluderede studier sjældent blev defineret. De foreslog at inkludere mere data om materialeegenskaber i bygningsobjekter, så BIM-data nemmere kan tilpasses LCA-datastrukturen. I forhold til dataudveksling mellem BIM-software og LCA-software anbefaler Soust-Verdaguer et al. at anvende openBIM, som IFC-filformatet, da det vil øge interoperabiliteten mellem værktøjer samt bryde med softwarespecifikke løsninger. Samme anbefaling gives af Santos et al. (2019), men det påpeges samtidig, at IFC-skemaet mangler brugbare IFC-egenskaber til miljøvurderinger.

Mangel på omfattende ontologier forhindrer automatisering, da klassifikation og indbyrdes relationer mellem objekter er nødvendige for automatisering. Santos et al. mener, at parametre bør tilføjes til bygningsobjekter, så de kan bruges som dataopbevaring og derved fremme overgangen fra informationsmodellering til informationshåndtering. Teng et al. (2022) påpeger, at interoperabiliteten mellem BIM-software og LCA-værktøjer er en af de mest signifikante udfordringer, fordi det er separate paradigmer iht. værktøjer, terminologi og datastruktur. LCA er en generisk metode, der er blevet udviklet til at kunne tilpasses enhver sektor, hvorimod BIM primært bruges i byggebranchen. Data indeholdt i BIM er derfor ikke tilstrækkeligt for LCA-værktøjer. Teng et al. foreslår, at mere information om materialeegenskaber tilkobles BIM, så BIM-databaser og LCA-databaser har en mere ensformig datastruktur. Automatisk dataindsamling opfordres ligeså, men det pointeres, at tidligere forsøg, såsom ens navngivning og individuelle ID, stadig krævede manuel databehandling.

3.5 Sammenstilling mellem teori og praksis

I dette afsnit opsummeres først de udfordringer, der blev identificeret ved at gennemgå studierne fra litteraturgrundlaget. Efterfølgende inddrages udfordringerne fra praksis, der blev fastslået under den indledende empiriindsamling. Til slut sammenstilles udfordringerne for at afgrænse, hvilke er mest værdiskabende at løse for brugeren, og det vurderes, hvorvidt udfordringerne tidsmæssigt er realistiske at løse. Afgrænsningen fungerer ligeledes som kontekstforståelse og specificering af forudsætninger til udviklingen af prototypen.

3.5.1 Opsummering af udfordringer

Studierne fra litteraturgrundlaget belyste adskillelige udfordringer relevant for BIM-LCA integrationen. I tabel 2 opsummeres og inddeles udfordringerne efter scenariet, de er gældende for (LCA, tidlig LCA eller begge). LCA-scenariet repræsenterer en situation, hvor formålet er at udarbejde en endelig LCA-beregning efter forskrifterne i bygningsreglementet. I tidlig LCA-scenariet er formålet at udarbejde en LCA-screening, der tidligt skal give en formodning om, hvorvidt den endelige LCA-beregning vil overholde grænseværdierne.

Gældende for	Udfordring
LCA	Manglende overensstemmelse i datastrukturen for IFC- og LCA-data.
LCA	LCA-egenskaber mangler i IFC-skemaet (udvidelse af IFC-skemaet påkræves).
LCA	Envejskommunikation fra BIM til LCA (eksisterende arbejde overskrives).
LCA	Manuel mapping for at tilknytte LCA-data til BIM-data.
LCA	Nødvendigt med genindtastning af data grundet datatab ved IFC-eksport.
LCA	Oprettelse af LCI er tidskrævende.
LCA	Tidskrævende at oprette Exchange Requirements og mappe dem til IFC-data.
Tidlig LCA	Forsimplede beregningskerner producerer uanvendelige LCA-beregninger.
Begge	Komponenttyper mangler i IFC-skemaet.
Begge	Manglende gennemsigtighed i LCA-beregningen samt resultaterne.
Begge	Virksomheder har forskellige behov til BIM-LCA workflows alt efter tilgængelige ressourcer og målsætninger.
Begge	Manglende muligheder for at sammenligne klimapåvirkningen i forskellige scenarier (materialevalg, bygningsdele).
Begge	LOD-niveauet mangler at blive specificeret i bygningsmodellerne (hvilke informationer der er behov for).

Tabel 2. Opsummering af udfordringer identificeret i litteraturgrundlaget.

Udfordringerne fra den indledende empiriindsamling er indhentet fra appendiks 1 samt opsummeret og ind-
delt i tabel 3.

Gældende for	Udfordring
LCA	Datagrundlaget er baseret på et udtræk fra bygningsmodellen. Ændringer i modellen opdateres først ved næste udtræk (ingen live-link).
LCA	Informationer tilført i LCAByg opdateres ikke automatisk i LCI'en.
LCA	Mængdeudtrækket (data fra Revit-skemaerne) kræver i nogle tilfælde en manuel bearbejdning, inden det indtastes i LCI'en.
Tidlig LCA	Materialevalget er fastlåst i de udvalgte generiske opbygninger.
Tidlig LCA	LCA-beregningsmetoden til det selvudviklede plug-in til Revit er ikke gyldig efter DS/EN 15978:2012.
Tidlig LCA	Data fra tidlige screeninger videreføres ikke, når en mere detaljeret LCA skal udarbejdes. Det er derfor nødvendigt at genindtaste data.
Begge	Der er ikke udarbejdet en kontrolproces for, om alle relevante objekter er indeholdt i modellen, eller om de har det nødvendige informationsniveau.

Tabel 3. Opsummering af udfordringer identificeret hos brugeren.

3.5.2 Sammenstilling og afgrænsning

Den første afgrænsning har at gøre med valget af LCA-scenarie, hvor det er valgt at fokusere på udarbejdelsen af en endelig LCA-beregning efter bestemmelserne i bygningsreglementet. I den tidlige LCA-screening anvendes ofte generisk LCA-data, da screeningen blot skal give en formodning om, hvorvidt grænseværdierne kan overholdes. Plug-in-løsninger i BIM-software kan hurtigere give feedback til brugeren, end hvis bygningsmodellen først skulle eksporteres til IFC. Integrationen mellem IFC-data og LCA-software er derfor mest værdifuld i den endelige LCA-beregning, da behovet for at specificere LCA-data er betydeligt større end ved en tidlig LCA-screening. Derudover har virksomheden allerede implementeret software, der varetager den tidlige LCA-screening. Udfordringerne gældende for tidlig LCA-scenariet i tabel 2 og 3 fravælges dermed at undersøges.

Grundet specialets IFC-fokus eksisterer en naturlig afgrænsning iht. Wastiels og Decuyperes (2019) integrationsstrategier (specificeret i afsnit 3.4.1), hvor LCA plug-ins i BIM-software udelukkes (strategi nummer 4). Eksport af BoQ fra BIM-software (strategi nummer 1) fravælges ligeledes, da IFC-skemaet kan rumme mængder og materialer fra bygningsobjekter. Prototypen skal importere relevant egenskabsdata fra IFC-objekter ind i LCA-software, hvilket er muligt med integrationsstrategi nummer 2, 3 og 5. Under den indledende empiriindsamling, i samtalen med LCA-specialisten, blev det nævnt, at det foretrækkes at arbejde med LCA-data

i LCA-software. Det ekskluderer strategi nummer 5, da bygningsobjekter beriges med LCA-data i BIM-software fremfor LCA-software. I strategi nummer 3 linkes LCA-data til bygningsobjekter igennem en BIM-viewer, således at LCA-data kan forblive i LCA-softwaremiljøet. At sammenkoble data mellem BIM-software, BIM-viewer og LCA-software vurderes dog som en meget tidskrævende proces, der er urealiserbar inden for speciallets tidsramme. I udviklingen af prototypen tages der derfor udgangspunkt i Wastiels og Decuyperes (2019) integrationsstrategi nummer 2, hvor bygningsobjekters geometriske parametre samt relevante egenskabsdata importeres ind i LCA-software ved brug af IFC.

En anden selvskreven afgrænsning er valget af LCA-software, da virksomheden anvender LCAbyg til at beregne deres klimapåvirkning. I det danske bygningsreglement gives der mulighed for at beregne klimapåvirkningen efter selvvalgt LCA-software, men det kræver, at softwaret udfører beregningen iht. DS/EN15978:2012 (Bygningsreglementet, 2018). LCAbyg er tilpasset de danske forhold og tillader tredjepartsintegration igennem JSON-formatet. Det er dermed sikret, at klimapåvirkningsberegningen udføres efter forskrifterne i lovgivningen, og at eksternt data kan tilføjes i softwaret.

Udover afgrænsningen af integrationsstrategi og valg af LCA-software kan fokusområder for udviklingen af prototypen identificeres ved at sammenstille udfordringerne fra teori og praksis. Et eksempel herpå er oprettelsen af en LCI, der i litteraturen blev identificeret som en tidskrævende proces. Det stemmer overens med, hvad der blev fundet i undersøgelsen af praksis, da data fra bygningsmodellen skulle igennem flere processer og programmer, før det kunne anvendes i LCAbyg. Først blev udvalgt egenskabsdata fra bygningsobjekter opsat i Revit, så dataet efterfølgende kunne eksporteres til Excel (BoQ). I Excel blev manuelle justeringer og opsummeringer af dataet udført, hvorefter det blev kopieret over i LCI'en, som afslutningsvis blev eksporteret til LCAbyg. I udviklingen af prototypen bør det derfor undersøges, hvordan oprettelsen af en LCI kan simplificeres, så processer mellemliggende BIM-software og LCA-software kan reduceres.

Et andet fællestræk mellem teori og praksis er den manglende kommunikation fra LCA-software tilbage til LCI'en (eller BIM-softwaret), der resulterer i manuelle genindtastninger og overskrivninger af eksisterende LCA-arbejde. Det skal derfor undersøges, hvordan data fra LCAbyg kan formidles tilbage til LCI'en, så brugeren ikke skal genindtaste alle data tilføjet i LCAbyg. Sideløbende skal det undersøges, hvordan brugerindtastede data i LCAbyg ikke overskrives, når bygningsmodeller opdateres med nye mængder eller objekter.

I litteraturgrundlaget blevet meget opmærksomhed rettet mod IFC-skemaet og dets mangler i LCA sammenhæng. Det omfatter alt fra forskellighed i datastruktur, manglende IFC-egenskaber og -komponenter, behov for at udvide IFC-skemaet og mapping af data. Disse udfordringer og mangler har ikke været mulige at bekræfte hos virksomheden, da IFC ikke anvendes i virksomhedens eksisterende LCA-proces. I udviklingen af

prototypen er det dermed især vigtigt at dokumentere IFC relaterede udfordringer, så validiteten og omfanget af udfordringerne kan præsenteres for virksomheden, såfremt virksomheden ønsker at ibrugtage IFC i deres LCA-proces.

Ved at sammenstille teori og praksis samt afgrænse omfanget af prototypeudviklingen kan følgende forudsætningerne opstilles:

- Prototypen skal muliggøre, at IFC-data fra bygningsobjekter kan anvendes i LCAbyg.
- Prototypen skal garantere, at information fra LCAbyg kan kommunikeres tilbage til LCI'en.
- Prototypen skal sikre, at IFC-modeller kan opdateres uden at overskrive brugerindtastede data i LCAbyg.
- Det skal undersøges, hvordan oprettelsen og opdateringer af en LCI kan forbedres.
- Det skal dokumenteres, hvilke problemer en IFC-LCA integration frembringer.

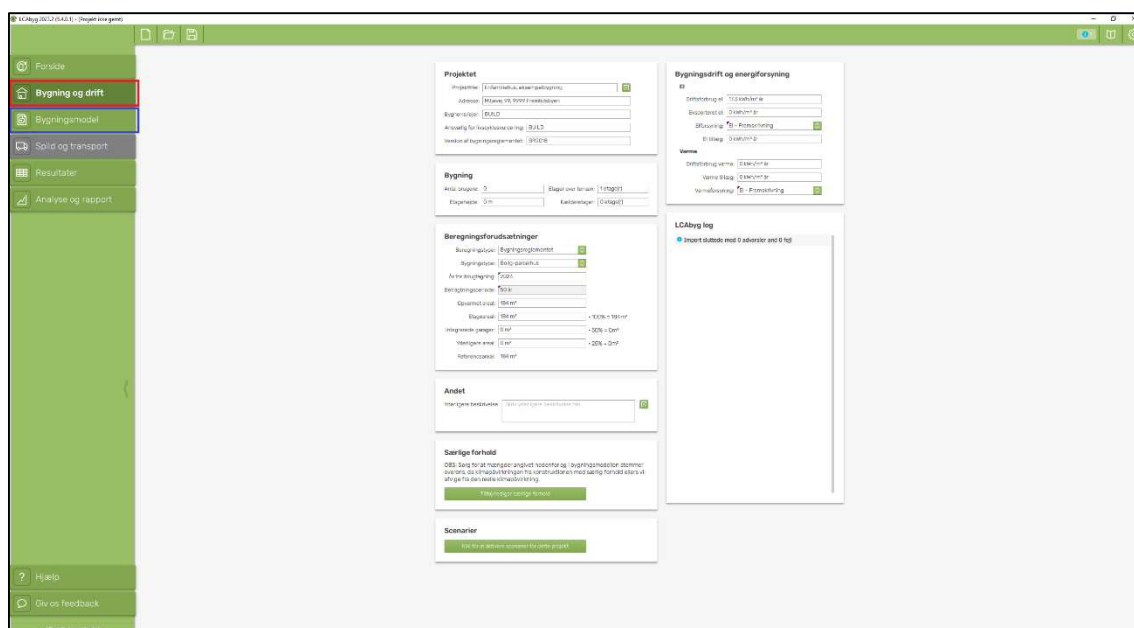
4. Udvikling af prototype

I afsnit 4.1 præsenteres LCA-softwaret LCAByg, så en grundlæggende forståelse af programmets datastruktur fås. Dernæst introduceres den indledende udvikling af prototypen, hvor udfordringer og muligheder belyses (afsnit 4.2). Omfanget af konceptdemonstrationen samt feedbacken derfra beskrives i afsnit 4.3, hvorefter der i afsnit 4.4 opfølges på demonstrationen. Afslutningsvist fremvises prototypen i afsnit 4.5.

4.1 Præsentation af LCAByg

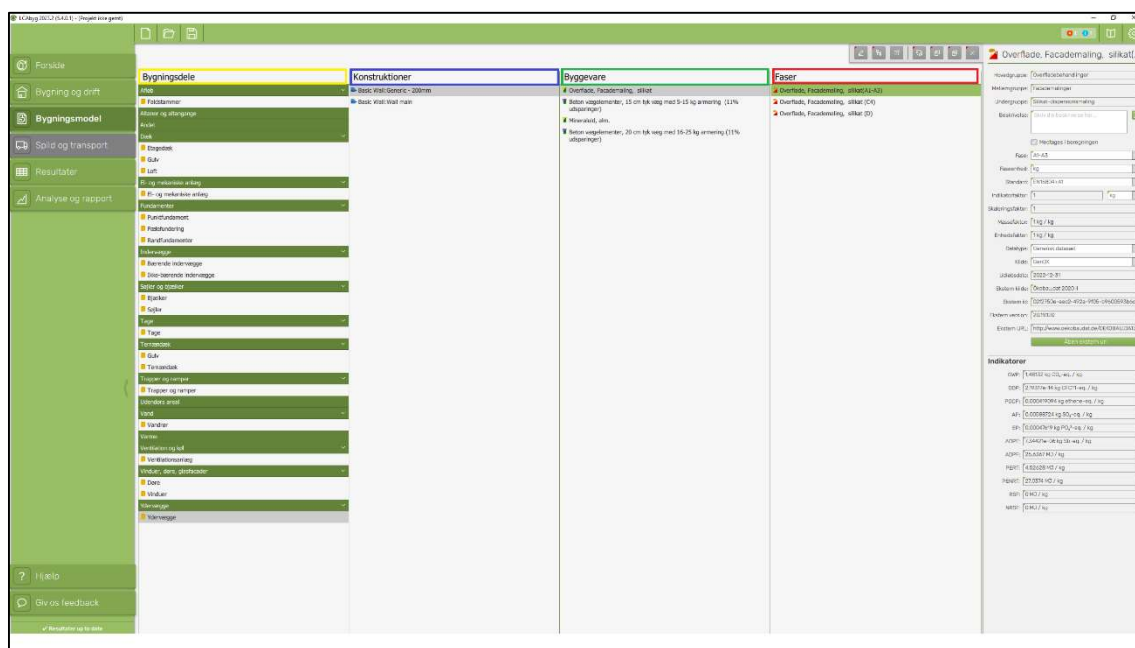
Det er vigtigt at forstå, hvordan LCAByg fungerer, inden udviklingen af de forskellige elementer i prototypen forklares, da hovedformålet for prototypen er at integrere data ind i LCAByg. I nedenstående afsnit gennemgås derfor LCABygs brugergrænseflade, samt hvordan den teknisk er opbygget.

I LCAByg er det muligt at tilføje alle informationer en LCA-beregning påkræver, men prototypen skal reducere mængden af informationer, der manuelt skal indtastes i programmet. Informationerne skal indtastes under to forskellige faner, henholdsvis "Bygning og drift" (driftsfanen) og "Bygningsmodel" (modelfanen). I figur 5 ses brugergrænsefladen for driftsfanen, hvor informationer om projektet, bygningen, beregningsforudsætningerne og driften skal indtastes. I prototypen er der udviklet en skabelon til driftsfanen, så informationer på forhold er indtastet, når et LCAByg-projekt åbnes. Ingen yderligere handlinger foretages under driftsfanen, så en dybdegående forklaring af fanen undlades. I stedet gennemgås modelfanen, da alt data fra bygningsmodellen opbevares herunder.



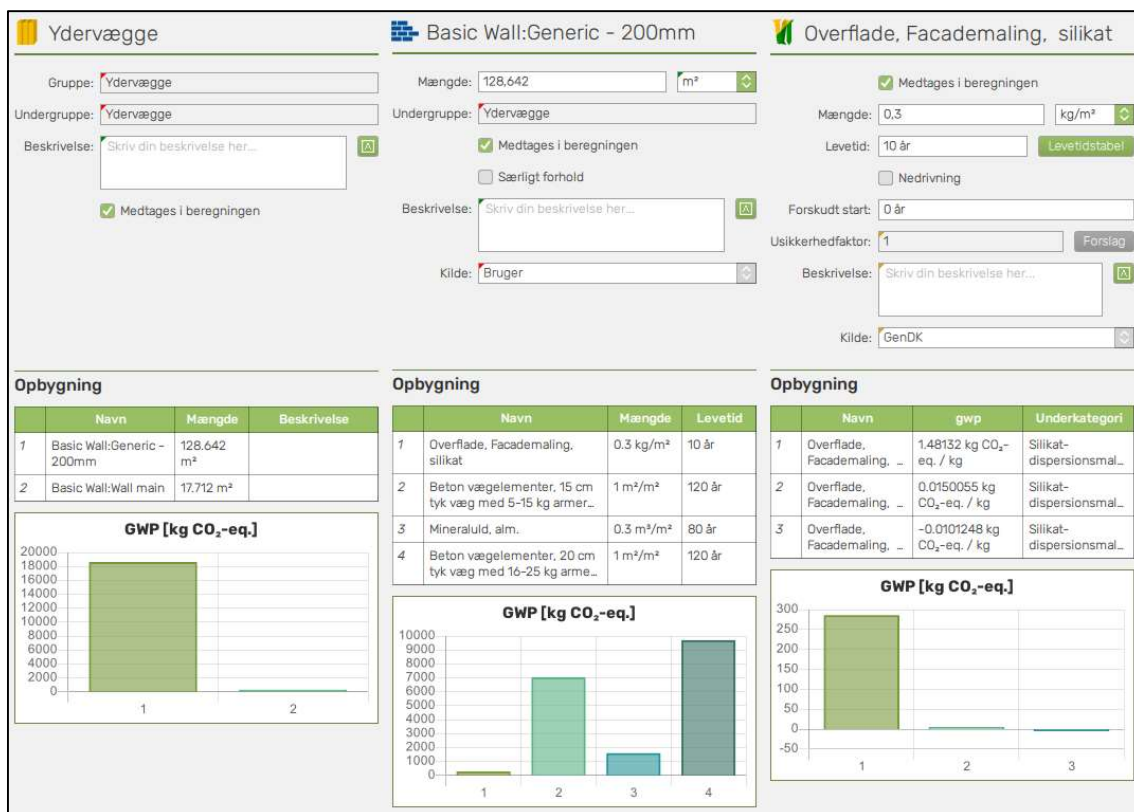
Figur 5. Brugergrænsefladen for driftsfanen (rød markering).

I figur 6 ses brugergrænsefladen for modelfanen, hvor data er struktureret i forskellige lag, hvilket er markeret med forskellige farver på figuren. Det første lag "Byggningsdele" er markeret med gul på figuren, og laget anvendes til at inddele bygningsobjekter efter deres objekttype. Det er nødvendigt at gøre, så LCAByg forstår, hvilken funktion bygningsobjekterne har (væg, vindue, dør, dæk, tag etc.). Det næste lag er "Konstruktioner" (blå markering), hvor alle bygningsobjekter af den valgte objekttype opbevares. Den grønne markering er det tredje lag "Byggevarer", der indeholder alle materialer, som bygningsobjektet består af. I det sidste lag "Faser" (rød markering) opbevares alt LCA-data relateret til det valgte materiale, hvilket er inddelt efter livscyklusstadier. I figur 6 kan det i højre side ses, hvilke data er tilknyttet det valgte livscyklusstadium.



Figur 6. Brugergrænsefladen for modelfanen.

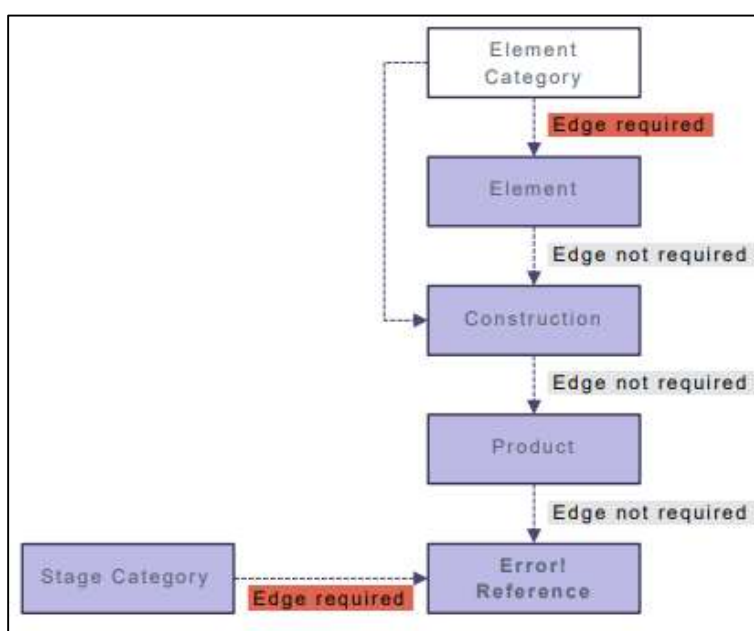
Det data, der vises længst til højre, afhænger af, hvad der i brugergrænsefladen er blevet valgt. Hvis der klikkes på et materiale, et bygningsobjekt eller en objekttype i stedet for et livscyklusstadium, så vil andet data visualiseres. I figur 7 er det visualiseret, hvordan det data til højre vil se ud, hvis en objekttype, et bygningsobjekt eller et materiale er valgt.



Figur 7. Data der visualiseres i LCAbyg ved valg af objekttype (venstre), bygningsobjekt (midt) eller materiale (højre).

Det data der vedrører materialer og deres livscyklusstadier er ikke noget, som prototypen eller IFC-modellen skal producere. Det er data fra LCA-databaser (f.eks. EPD Danmark) som importeres ind i LCAbyg. Data som vedrører objekttyper og bygningsobjekter er det, som prototypen skal importere ind i LCAbyg, hvilket er muligt ved at bruge JSON-formatet.

I figur 8 er det vist, hvordan data skal struktureres, så det kan importeres ind i LCAByg under modelfanen. Den hvide boks "Element Category" er fastlagte objekttypekategorier, som de tidligere beskrevet objekttyper (Bygningsdele) inddeles efter. De lilla bokse "Element", "Construction", "Product" og "Stage Category" er henholdsvis objekttyper, bygningsobjekter (Konstruktioner), materialer (Byggevare) og livscyklus stadier (Faser). Det er nødvendigt at lave en kobling (Edge) fra objekttypekategorierne til objekttyper og bygningsobjekter, så LCAByg kan inddele data korrekt. Til koblingen bruges UUID4 (Universally Unique Identifier), hvilket er en 36 karakters alfanumerisk id-kode. Alle elementer får tildelt en unik id-kode, som der kan refereres til i JSON-filerne. På den måde holder LCAByg styr på, hvad der er hvad i JSON-filerne. Når prototypen skal udtrække data fra en IFC-model, er det denne datastruktur, der skal struktureres og opbygges efter.



Figur 8. Illustration af LCABygs datastruktur (Bruhn & Sørensen, 2020).

4.2 Udvikling af nyt workflow

I dette afsnit startes beskrivelsen af prototypeudviklingen som indledes med at beskrive de tidlige overvejelser gjort og valg foretaget. Udfordringer opstået undervejs i processen belyses, og en udarbejdet konceptmodel fremvises afslutningsvist.

En af forudsætningerne for prototypen var at undersøge, hvordan oprettelsen af en LCI kunne forbedres. Det var dog nødvendigt først at finde ud af, hvilke informationer skulle være til stede i en LCI, så data kunne åbnes i LCAByg. I Bruhn og Sørensen (2020) JSON guide for LCAByg beskriver de, hvilke data er nødvendige i JSON-filerne, men ikke hvordan det data flyttes fra LCI'en til JSON-filerne. Første delmål for prototypeudviklingen var derfor at identificere, hvordan data kunne opstilles, så det var transparent for brugeren samtidig med, at det kunne bruges som input til JSON-filerne. Yderligere blev der taget en beslutning om, ikke at anvende softwarespecifikke filformater.

Igennem diverse afprøvninger af forskellige muligheder lykkedes det at finde en effektiv fremgangsmåde. CSV-formatet (comma-separated values) blev valgt som dataformat til LCI'en, da det tillader tabelformet data. Alle nødvendige data i LCI'en kunne dermed skrives på samme linje, så hver linje kunne repræsentere et bygningsobjekt. Endvidere blev det valgt at programmere i programmeringssproget Python, da det fungerer godt med CSV- og JSON-formatet. PyCharm blev valgt som det integrerede softwareudviklingsmiljø, som programmeringen skulle foregå i. Herudover blev det besluttet at anvende udvidelsespakker til Python, da de giver gode muligheder for at modificere og visualisere data. Det var nødvendigt at bruge 13 parametre i LCI'en for at rumme alt data og sikre, at det kunne integreres ind i LCAByg (uddybes i afsnit 5.1).

Prototypens næste delmål var at finde ud af, hvordan data fra IFC-modellen kunne udtrækkes og indsættes i LCI'en. Det blev gjort ved at undersøge, hvilke egenskabsdata der er tilgængelige i de forskellige versioner af IFC-skemaet og MVD'er.

Bygningsobjekter er indeholdt i kerne-laget (Core layer) af IFC-skemaet, hvor alle entiteter har et GUID (globally unique id), der kan bruges til at identificere objekterne (buildingSMART, 2017). Alt efter versionen af IFC-skemaet er der dog stor forskel på, hvilke andre data tilknyttes bygningsobjekter, når de eksporteres fra BIM-software til IFC. To versioner af IFC-skemaet blev undersøgt, da de fleste MVD'er i Revit er baseret på disse versioner. Det er version IFC2X3 TC1 og IFC4 ADD2 TC1, der begge er officielle ISO-standarder (ISO/PAS 16739:2005 og ISO 16739-1:2018). Den største forskel, for prototypen, var, at bygningsobjekter i IFC2X3 ikke blev eksporteret med tilhørende mængder (Base Quantities). Det gjorde de derimod i IFC4, hvilket blev årsagen til, at IFC-modellerne, som prototypen skulle udtrække data fra, skulle være eksporteret til version IFC4 ADD2 TC1.

Undervejs i prototypeudviklingen blev adskillige IFC relaterede udfordringer noteret. Det var især bygningsobjekternes IFC-klassificeringer, der skabte problemer. Det der menes med IFC-klassificeringer er, hvilken undertype af `IfcBuildingElement` bygningsobjekterne i BIM-software bliver eksporteret som. I tabel 4 ses de 21 forskellige undertyper af `IfcBuildingElement`.

IfcBuildingElement		
IfcBeam	IfcFooting	IfcRoof
IfcBuildingElementProxy	IfcMember	IfcShadingDevice
IfcChimney	IfcPile	IfcSlab
IfcColumn	IfcPlate	IfcStair
IfcCovering	IfcRailing	IfcStairFlight
IfcCurtainWall	IfcRamp	IfcWall
IfcDoor	IfcRampFlight	IfcWindow

Tabel 4. Undertyper af `IfcBuildingElement`.

`IfcBuildingElementProxy` kan bruges som stedfortræder af en undertype af `IfcBuildingElement`, da den kan indeholde samme funktioner som undertyperne, men uden at specificere hvilket bygningsobjekt det repræsenterer (buildingSMART, 2017). Det vil sige, at bygningsobjekter uden en prædefineret undertype vil blive defineret som `IfcBuildingElementProxy`, når en bygningsmodel bliver eksporteret til IFC. Det stiller et krav til bygningsmodellen om, at bygningsobjekter skal være klassificeret korrekt inden eksport. Hvis et bygningsobjekt bliver eksporteret som `IfcBuildingElementProxy`, så skyldes det manglende klassificering i bygningsmodellen. I udviklingen af prototypen blev det identificeret, at det ofte var et problem ved sammensatte bygningsobjekter. I BIM-softwaret blev det sammensatte bygningsobjekt betragtet som et enkeltstående element, men når bygningsmodellen blev eksporteret til IFC, så blev objektet eksporteret som adskillige bygningsobjekter. Det blev derfor besluttet, at prototypen ikke skulle håndtere bygningsobjekter, som eksporteret som `IfcBuildingElementProxy`.

Samme eksportudfordringer blev fundet ved brugen af `IfcCurtainWall`, der repræsenterer en ydervæg i en bygning som er: "[...] an assembly of components, hung from the edge of the floor/roof structure rather than bearing on a floor" (buildingSMART, 2017). I bygningsmodeller bliver eksempelvis glasfacadesystemer eksporteret som `IfcCurtainWall`, men de eksporteres ikke som et samlet system. Hver komponent i glasfacadesystemet eksporteres som et enkeltstående bygningsobjekt, hvoraf `IfcPlate` repræsenterer glasarealet, og `IfcMember` bruges til stolperne/sprosserne mellem glasarealet (kaldes mullions i Revit). I figur 9 ses et eksempel på, hvilke data er tilgængelige fra et `IfcCurtainWall`-objekt.

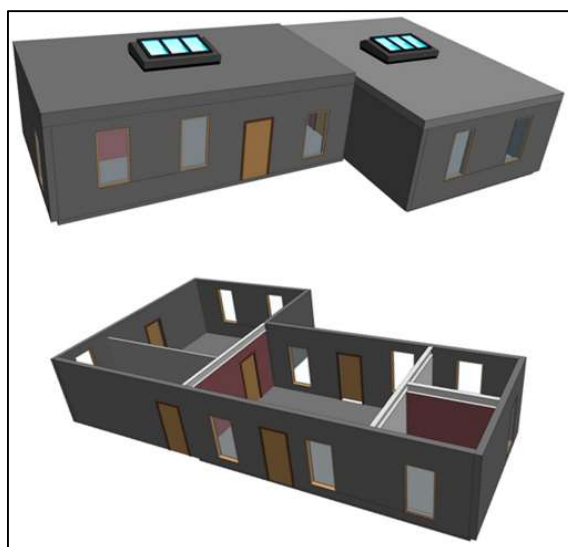
```
#326=IfcCurtainWall('2HAAFR_0H52B9xUjEj$mFy',#18,'Curtain Wall:Exterior Glazing:350459',
$, 'Curtain Wall:Exterior Glazing',#325,$, '350459', .NOTDEFINED.)
{'IsExternal': True, 'id': 722, 'Reference': 'Exterior Glazing'}
{'Reference': 'Exterior Glazing', 'id': 723}
{'Length': 6800.000000000002, 'id': 810}
```

Figur 9. Dataudtræk fra et bygningsobjekt klassificeret som IfcCurtainWall.

Det eneste mængde-parameter er længden på glasfacadesystemet, hvilket er ubrugeligt uden højden. Det kræver derfor, at mængdedata udtrækkes fra IfcMember- og IfcPlate-objekter. For objekter klassificerede som IfcMember kan både længden, arealet og volumen udtrækkes. For objekter klassificerede som IfcPlate burde bredden, arealet og volumen, ifølge dokumentationen for IFC4_ADD2_TC1, kunne udtrækkes. Realiteten er dog en anden, når glasfacadesystemer eksporteres fra Revit til IFC, hvor kun bredden og volumen udtrækkes. Arealet, der kan argumenteres for, er det vigtigste mængde-parameter, mangler. Af den årsag blev det vurderet, at bygningsobjekter klassificerede som IfcCurtainWall, IfcMember og IfcPlate udeblev fra prototypen i første omgang.

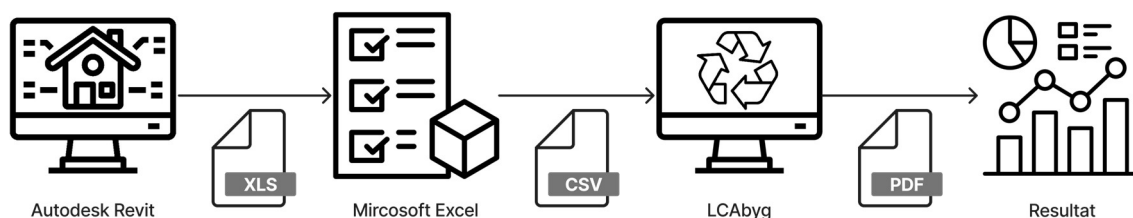
Af tidsmæssige årsager blev det besluttet, at prototypen blot skulle håndtere de mest kritiske undertyper af IfcBuildingElement relevant for LCA-beregningen. Det resulterede i, at prototypen kan håndtere følgende undertyper: IfcBeam, IfcColumn, IfcDoor, IfcRoof, IfcSlab, IfcWall og IfcWindow. De resterende 14 undertyper er prototypen dermed ikke i stand til at håndtere på nuværende tidspunkt.

Til at afprøve prototypen blev en konceptmodel modelleret i Revit, hvorfra den blev eksporteret til IFC ved brug af buildingSMARTs "Reference View" MVD (opsat for arkitektmodeller), der benytter IFC4 ADD2 TC1-skemaet. Konceptmodellen består af 34 bygningsobjekter, som alle kan håndteres af prototypen. I figur 10 er konceptmodellen visualiseret.



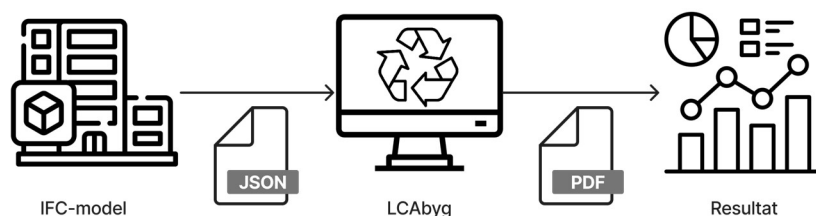
Figur 10. Visualisering af konceptmodel.

Inden konceptdemonstrationen hos brugeren fandt sted, var det lykkedes at udarbejde et nyt workflow, så det kunne visualiseres for brugeren. Brugers eksisterende workflow er visualiseret i figur 11, hvilket blev fastlagt under undersøgelsen af brugers arbejdsprocesser (appendiks 1).



Figur 11. Illustration af brugers eksisterende workflow.

Det nye workflow, der blev præsenteret for brugeren, kan ses i figur 12.



Figur 12. Illustration af nyt workflow.

Prototypen var på daværende tidspunkt i stand til at udtrække mængder fra bygningsobjekterne i konceptmodellen og oprette de nødvendige JSON-filer, så objekterne og deres mængder kunne tilgås i LCAByg.

4.3 Konceptdemonstration

Prototypen blev, hos brugeren, præsenteret for to LCA-specialister, to BIM-specialister samt en bygningskonstruktør. Efter demonstrationen blev der indledt i en dialog om, hvordan prototypen kunne videreudvikles, samt hvilke funktioner den skulle understøtte. Kommentarerne fra demonstrationen og dialogen præsenteres i dette afsnit, og opsummeres i tabel 5.

Der blev set et stort potentiale i det nye workflow, da det ville reducere flere manuelle processer fra det eksisterende. Mange funktioner blev dog udpeget til videreudviklingen, hvoraf kommunikation fra LCAByg tilbage til LCI'en var øverst på listen. Tovejskommunikation ville eliminere genindtastninger, når data tilført i LCAByg skulle opdateres i LCI'en. Det vil dermed være en stor forbedring af workflowet. Hertil blev det foreslået, at man kunne undersøge mulighederne for at oprette en LCI online, så data fra LCI'en kunne anvendes til yderligere formål end LCA-beregningen. Det blev af BIM-specialisterne påpeget, at en online adgang ville tillade flere personer at tilgå og ændre i LCI'en. Et forslag der delte vandene, da LCA-specialisterne foretrak soloadgang. Online adgang til prototypen, så den nemt kunne anvendes uden kendskab til programmering, blev derimod vurderet som nyttigt af begge parter.

Anvendelsen af IFC4 ADD2 TC1-skemaet fandt ingen begejstring blandt BIM-specialisterne, da det nærmest aldrig blev brugt på byggeprojekter. I stedet blev det foreslået at anvende IFC2X3 TC1-skemaet, hvor IFC-eksporten fra BIM-softwaren modificeres, så mængder fra bygningsobjekter (Base Quantities) bliver eksporteret. Endvidere var det nødvendigt at prototypen kunne tilføje bygningsobjekter, der ikke er indeholdt i bygningsmodellen, da eksternt data fra samarbejdspartnere sommetider kunne være forsinket.

Der blev givet få rettelser til de eksisterende funktioner i prototypen, men placeringen af data i LCAByg var nødvendig at ændre, så den stemte overens med brugerens dataopbygning. Prototypen importerede bygningsobjekterne fra konceptmodellen til de fastelaste objekttypekategorier i LCAByg. I stedet skulle prototypen først oprette objekttyper (Bygningsdele) under de fastlagte objekttypekategorier. Herefter skulle bygningsobjekterne importeres til de nye objekttyper.

Det sidste som blev diskuteret var, hvorledes bygningsobjekter skulle prædefineres med materialer, når de blev importeret til LCAByg. Argumentet for var, at LCA-specialisterne undgik at bruge tid på at tilknytte materialer til bygningsobjekter i LCAByg. Argumentet imod var, at LCA-specialisterne havde brug for at kontrollere, hvad data blev tilknyttet hvilke bygningsobjekter. I prototypen vil en funktion, der tilknytter materialer til bygningsobjekter, relativt hurtigt kunne tilføjes. Udfordringen er at definere et standardiseret grundlag for, hvilke materialer prædefineres for hvilke objekttyper. Endvidere vil en LCA-specialist skulle kontrollere materialeopbygningerne på et tidspunkt, hvilket kan ende med at være mere tidskrævende i det lange løb.

Det blev foreslået at basere prædefineringen på klassifikationskoder, så et mere detaljeret udgangspunkt end IFC-objekttypen kunne anvendes. Selvom klassifikationskoder er et bedre udgangspunkt, fortæller de dog ikke noget om materialeopbygningen. Det vil være nødvendigt at detaljere klassifikationskoderne til et ekstremt niveau, hvilket vil være særledes tidskrævende og omstændigt.

Essentielle funktioner	Potentielle funktioner
Tovejskommunikation mellem LCAbyg og LCI.	Online LCI.
Udskift IFC4 ADD2 TC1 med IFC2x3 TC1.	Online adgang til prototype.
Tilføj bygningsobjekter fraværende i IFC-model.	Prædefinering af materialer.
Tilpas dataopbygning i LCAbyg.	

Tabel 5. Opsummering fra konceptdemonstration.

4.4 Opfølgning på konceptdemonstration

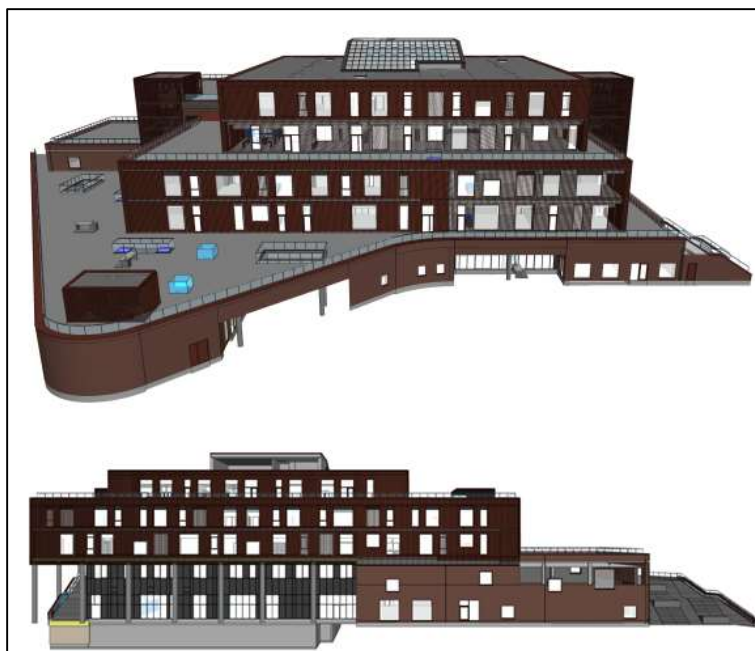
Kommunikationen mellem LCAbyg og LCI'en blev, på baggrund af konceptdemonstrationen, vurderet som den vigtigste funktion at inddrage i prototypen. Det var dermed nødvendigt at identificere, hvordan data tilført i LCAbyg blev opbevaret i JSON-filerne, da prototypen skulle udtrække data fra LCA-projektet fremfor at importere data til projektet. Adskillige fremgangsmåder blev afprøvet, så det kunne sikres, at alt nyt og eksisterende data blev bibeholdt. I afsnit 5.2 findes en teknisk gennemgang af, hvordan prototypen lykkedes med at skabe tovejskommunikationen.

Det valgte IFC4-skema blev uden problemer udskiftet med IFC2X3 TC1-skemaet, da samme mængde-data var tilgængelige, såfremt IFC-eksporten var opsat korrekt i BIM-softwaren. Yderligere blev enkelte tekniske justeringer til den eksisterende Python-kode foretaget, da det efter demonstrationen blev bemærket, at bygningsobjekterne havde de forkerte mængder i LCAbyg. Det skyldtes manglende sortering af bygningsobjekterne samt konvertering fra millimeter til meter. Muligheden for at anvende XLS-formatet (Microsoft Excel) fremfor CSV-formatet blev også tilføjet til prototypen.

Muligheden for at tilføje bygningsobjekter, der ikke er indeholdt i IFC-modellen, blev ligeledes inkorporeret i prototypen. Herudover blev det gjort muligt at opdatere IFC-modellen uden at overskrive eksisterende data. Det blev testet på en opdateret version af konceptmodellen, hvor nye bygningsobjekter var tilføjet, og eksisterende objekter var ændret eller fjernet. Den sidste funktion, der blev tilføjet til prototypen, var at oprette et LCAbyg-projekt ud fra en LCI fremfor en IFC-model. Alle nye funktioner på nær sidstnævnte vil være en del af brugerens workflow, når der skal udarbejdes en LCA-beregning. Muligheden for at undgå IFC-tilgangen blev tilføjet, så brugeren kan udskifte deres eksisterende workflow til prototypens med det samme. Det er endnu ikke muligt med IFC-tilgangen, da ikke alle IFC-objekttyper kan håndteres.

Prototypeversionen, som blev fremvist til konceptdemonstrationen, bestod af et langt Python-script, men fordi de nye funktioner skulle tilføjes, krævede det, at flere Python-scripts skulle udarbejdes. Det blev identificeret, at fem scripts skulle udarbejdes for at understøtte alle prototypens funktioner, hvilket uddybes i afsnit 4.5.

Til konceptdemonstrationen blev det i en sidebemærkning nævnt, at det kunne være spændende at teste prototypen på en mere virkelighedsnær bygningsmodel. Brugeren tilbød derfor at tilvejebringe en bygningsmodel fra et tidligere byggeprojekt, der var markant større end konceptmodellen. Brugers bygningsmodel indeholder 3904 bygningsobjekter, som prototypen er kompatibel med, og kan visualiseres i figur 13.



Figur 13. Rumlig afbildning af brugerens bygningsmodel.

Prototypen blev afprøvet på den nye bygningsmodel, hvorved nye udfordringer blev identificeret. Den største udfordring relaterede sig til vægobjekterne, når de skal eksporteres fra BIM-software til IFC. I eksporten kan vægobjekterne enten klassificeres som `IfcWall` eller `IfcWallStandardCase` (buildingSMART, 2007). Sidstnævnte er en undertype af `IfcWall`, men den skaber ikke udfordringer. Ud af 2043 vægobjekter blev 93,78 % klassificerede som `IfcWallStandardCase` og resten som `IfcWall` (127 stk.). Alle `IfcWall`-objekter har ingen mængder tilknyttet, hvilket betyder, at de skaber problemer med LCA-beregningen. Det er en IFC relateret udfordring, da mængderne fremgår af mængdeudtrækket foretaget i BIM-softwaret. Det har ikke været muligt at finde en officiel forklaring på, hvorfor vægobjekterne klassificeres som `IfcWall` fremfor `IfcWallStandardCase`. I undersøgelsen af udfordringen er det dog bemærket, at alle vægobjekter med en smule kompleksitet, såsom at dele af væggen er afskåret, eller at væggen fastgjort til en skrå overflade, klassificeres som `IfcWall`. Hvis data fra vægobjekter i IFC-modeller skal være pålidelige, kræver det derfor, at der stilles krav til modelleringsteknikken.

En anden udfordring opstod, når tagobjekter blev klassificerede som `IfcSlab`-objekter. Selvom `IfcSlab`-objekterne havde en prædefineret type som tag, så gav det udfordringer i prototypen. I PyCharm, det integrerede softwaremiljø som prototypen udarbejdes i, blev udvidelsespakken `Ifcopenshell` anvendt til at udtrække data fra IFC-modellerne. Når tagobjekterne klassificerede som `IfcSlab`-objekter skulle udtrækkes, så blev de både udtrukket som `IfcSlab`- og `IfcRoof`-objekter. Det medførte, at tagobjekterne indgik dobbelt i prototypen. Udfordringen blev løst ved at frasortere alle `IfcSlab`-objekterne fra prototypen, så kun `IfcRoof`-objekterne

blev udtrukket. Løsningen er ikke helt problemfri, da alle IfcRoof-objekter ikke besidder Base Quantities. Det er muligt at udtrække arealet af tagobjektet igennem IFC-egenskabssættet "Pset_RoofCommon", hvor det projekterede og total arealet kan udtrækkes. Volumen af tagobjekterne kan dog ikke udtrækkes og forbliver en mangel i prototypen.

Den sidste ændring i prototypen var at gøre den kompatibel med IfcCurtainWall-objekter, men fordi mængder ikke kunne udtrækkes, indgik objekterne i LCI'en uden mængder. Samme løsning blev brugt til IfcWall-objekter og et IfcRoof-objekt, modelleret som et ovenlysvindue-system. Det er dog muligt for brugeren manuelt at indtaste de manglende mængder i LCI'en, så de indgår korrekt i LCAbyg-projektet. Grundet disse ændringer var det muligt at anvende prototypen på brugerens bygningsmodel, uden problemer opstod.

4.5 Præsentation af prototype

Prototypen består af fem Python-scripts, der hver understøtter en bestemt funktion i brugerens LCA-workflow. En kort funktionsbeskrivelse af hvert script fås i tabel 6.

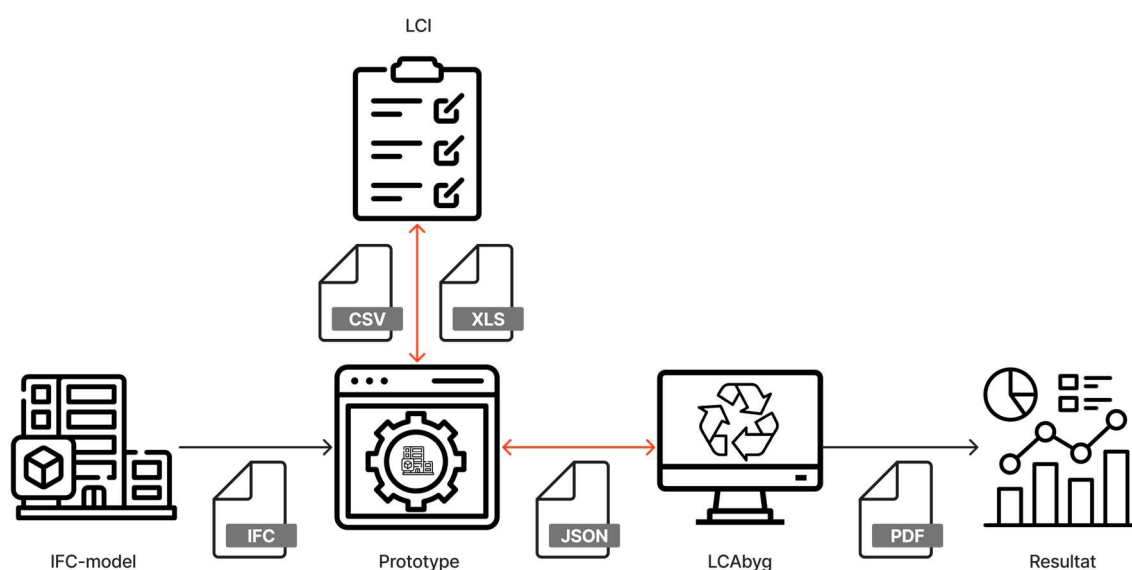
Navn	Funktion
Create from IFC	Opretter LCI og LCAbyg-projekt med bygningsobjekterne fra en IFC-model.
Update from LCAbyg	Opdaterer LCI med data tilført i LCAbyg.
Update from IFC	Opdaterer LCI og LCAbyg-projekt med data fra ny version af IFC-modellen.
Create from LCI	Opretter nyt LCAbyg-projekt med bygningsobjekterne fra en LCI.
Update from LCI	Opdaterer LCAbyg-projekt med manuelle indtastede data fra LCI'en.

Tabel 6. Navngivning af scripts samt kort funktionsbeskrivelse.

Programmeringen af de fem scripts kan ses i appendiks 3, hvor alle scripts er samlet. På nuværende tidspunkt er der ikke udviklet en brugergrænseflade til prototypen, men de fem scripts kan eksekveres fra PyCharm (eller andre integrerede softwareudviklingsmiljøer).

Prototypen er i stand til at håndtere følgende undertyper af IfcBuildingElement: IfcBeam, IfcColumn, IfcCurtainWall, IfcDoor, IfcRoof, IfcSlab, IfcWall og IfcWindow. De resterende undertyper kan tilføjes på et senere tidspunkt, da prototypen er opsat til hurtigt at kunne udvides.

I figur 14 er det illustreret, hvordan prototypen faciliterer et nyt workflow for brugeren til udarbejdelsen af LCA-beregninger. De røde pile på illustrationen repræsenterer tovejskommunikation, hvilket er gældende mellem prototypen, LCI'en og LCAbyg. Illustrationen kan bruges til at visualisere, hvilke funktioner der understøttes af prototypen, når den teknisk gennemgås i afsnit 5.



Figur 14. Illustration af brugerens nye workflow med prototypen.

5. Teknisk gennemgang af prototype

I nedenstående underkapitler gives en teknisk beskrivelse af, hvad der konkret sker i hvert af de fem udarbejdede Python-scripts. De gennemgås i den rækkefølge, som de er udset at skulle eksekveres i, hvorigennem det forventes at fremhæve brugerens nye workflow. For hvert script introduceres det indledningsvist, hvad der overordnet sker i scriptet, hvorpå en mere detaljeret gennemgang udføres.

5.1 Create from IFC

Det første script anvendes til at oprette et projekt i LCAbyg ud fra en IFC-model. Først udtrækkes objekter og deres mængder fra en IFC-model, hvorefter objekterne inddeles efter objekttype og mængderne opsummeres efter objekttypenavn. Efterfølgende indsættes objekttypenavnene og deres tilhørende mængder i en ny-oprettet LCI. Hvert typenavn i LCI'en bliver tildelt adskillige id, der bruges til at oprette JSON-filer, så informationerne i LCI'en kan overføres til LCAbyg. Afslutningsvis oprettes de resterende JSON-filer, der er nødvendige for, at et projekt kan åbnes i LCAbyg igennem JSON-integrationen.

Detaljeret gennemgang

I starten af scriptet importeres de udvidelsespakker som bruges undervejs, hvilke er: os, ifcopenshell, pandas, json og uuid. Herefter opsættes scriptet med den ønskede sti til projektfolderen, lokationen for IFC-modellen og enheder for bygningsobjekttyperne vælges. Opsætningen skal manuelt skrives ind i scriptet, men når en brugergrænseflade bliver udviklet, kan det ændres til brugervenlige inputs. Resten af scriptet kræver ingen yderligere indblanding fra brugeren.

```
9  # SET UP SCRIPT
10 Project_folder = "C:/Users/Krogh/Desktop/Virksomhed/"
11 Location_IFC_model = "C:/Users/Krogh/Desktop/Model_virksomhed.ifc"
12
13 # CHOOSE UNIT (M,M2,M3,Pcs)
14 Unit_IfcBeam_list = "M3"
15 Unit_IfcColumn_list = "M3"
16 Unit_IfcDoor_list = "Pcs"
17 Unit_IfcRoof_list = "M2"
18 Unit_IfcSlab_list = "M2"
19 Unit_IfcWall_list = "M2"
20 Unit_IfcWindow_list = "M2"
```

Figur 15. Udklip fra PyCharm af scriptets opsætning.

Efter opsætningen indhentes alle IfcBuildingElement fra IFC-modellen, hvor de IFC-objekter prototypen ikke kan håndtere frasorteres. Inden data udtrækkes fra IFC-objekterne åbnes to JSON-filer, der er importeret ind

i det integrerede softwareudviklingsmiljø, da de fungerer som en skabelon for struktureringen i LCAByg. Struktureringen kan modificeres, hvis det ønskes, men prototypens nuværende strukturering opretter 21 bygningsdele i LCAByg, som IFC-objekterne kan fordeles ud på.

Næste trin i scriptet er at udtrække data fra alle IFC-objekterne, der senere skal bruges til at oprette LCI'en. En Excel- og CSV-fil, der indeholder alt det valgte data fra IFC-objekterne, bliver eksporteret til projektfolder. Det er valgt at udtrække objekternes IFC-type, typenavn, mængde og GUID, da disse parametre er nødvendige for scriptet og LCA-beregningen. Yderligere tilføres hvert objekt en LCAByg-kategori (baseret på deres IFC-type) og LCAByg-id samt enhed (valgt i opsætningen).

IFC_entity	LCA_category	Type_name	Amount	Unit	GUID	LCA_category_id
IfcBeam	Bjælker	CW_Beam_Steel-H:Beam_Steel_HE140B	0,028	M3	2QKSL\$EW1A1x1d11jzD1	ec9ee040-9c1d-4cae-864c-4c6a0e4b8c5b
IfcDoor	Døre	311 - UD.V. INDGANGSDØR, ALU/ALU CW DOBBELT UDENPÅLIGGENDE SK	1	Pcs	00dxKXpHX5AhP_5vjoZahW	9bd14e05-5fba-4ecb-ab41-7940a8a0dbdc
IfcSlab	Gulv	Floor:131-001 - TERRÆNDEK, BUND DOBBELTKONSTRUKTION	1631,157	M2	2802QR5rP2JuBryQkseQ8	bf2534e2-87e6-40cb-9fff-17fb3d85a52a
IfcWallStandardCase	Ikke-bærende indervægge	Basic Wall:211-004 - UD.V. SANDWICHELEMENT, 650MM	21,973	M2	24UytbkC6XRYTGsN7QX9W	a9ab6709-61e1-46c1-a2e8-df00bdd0bb91
IfcCurtainWall	Ikke-bærende indervægge	Curtain Wall:226-001 - IND.V. GLASSYSTEMVÆGGE	0	M2	1qGcdS4mtSTA7iLHdpmF6	a9ab6709-61e1-46c1-a2e8-df00bdd0bb91
IfcWall	Ikke-bærende indervægge	VINDSKEDE MURKRONE AFDÆKNING (SANDWICH) OVENLYS:317-005 - VII	0	M2	1Xoho3cnf0MgKcHAVnEUET	a9ab6709-61e1-46c1-a2e8-df00bdd0bb91
IfcColumn	Søjler	CW_Column_Concrete-Prefab:Column_Concrete-Prefab_360x360	0,478	M3	10aVL1GuxOLRBuZMotZc5N	506a420e-c3cd-4849-a759-3117f10937b1
IfcRoof	Tag	Basic Roof:273-001 - VARMTTAG - TAG	1587,245	M2	2eVCgn_pb8F9GN4LUsswoq	d734712a-d27d-42c5-936f-98fe4c4df90b
IfcWindow	Vinduer	31 - Vinduesfals:319-005 - BETONFALS 210MM	5,969	M2	3W733Sy94Q8Gf4YQQSif	dcbf55db-3189-4d22-9b2b-2e27c057b363

Figur 16. Udsnit fra Excel-filen der gemmes i projektfolderen.

Når scriptet har eksporteret Excel- og CSV-filen til projektfolderen påbegyndes udarbejdelsen af LCI'en, hvor data skal tilføjes til 13 parametre, eksemplificeret i figur 17 og 18. IFC-typen og GUID er ikke relevante parametre at have med i LCI'en, men de er nødvendige for at identificere ændringer, når bygningsmodellen opdateres (Update from IFC). De resterende data skal overflyttes til LCI'en, men mængderne skal først opsummeres inden dataet overflyttes, da kun den samlede mængde for hvert typenavn skal bruges i LCA-beregningen.

Bygningsdel	LCA_category	Type_name	Amount	Unit	lifespan	Delayed_start	Construction_id
Dæk	Gulv	Floor:131-001 - TERRÆNDEK, BUND DOBBELTKONSTRUKTION	5054,553	M2	-	-	93f521bc-750b-4f38-b4ed-ef0e39c9b9ef
Indervægge	Ikke-bærende indervægge	Basic Wall:211-004 - UD.V. SANDWICHELEMENT, 650MM	21,973	M2	-	-	391d1ab7-e5f0-4efb-a94e-bc114713b631
Søjler og bjælker	Bjælker	CW_Beam_Steel-H:Beam_Steel_HE140B	0,032	M3	-	-	2571555d-4ea1-439a-99fa-4ec1c8c7e038
Søjler og bjælker	Søjler	CW_Column_Concrete-Prefab:Column_Concrete-Prefab_360x360	0,478	M3	-	-	fe65bc58-7978-4b53-b5e3-07372dae8610
Tag	Tag	Basic Roof:273-001 - VARMTTAG - TAG	1765,182	M2	-	-	5bd7761d-0cd1-4f30-b902-c9ec30fa4d1f
Vinduer, døre, glasfacader	Døre	311 - UD.V. INDGANGSDØR, ALU/ALU CW DOBBELT UDENPÅLIGGENDE SKYD	3	Pcs	-	-	f4891663-fcc2-4499-9919-71381d9e3444
Vinduer, døre, glasfacader	Vinduer	312 - UD.V. VINDUE:312-001 - UD.V. VINDUE - 1230X1032	21,888	M2	-	-	5bd4b0ed-8313-4cef-aefb-2d5a59c70b20
Ydervægge	Ydervægge	Basic Wall:211-001 - UD.V. SANDWICHELEMENT, 520 MM	1826,865	M2	-	-	69227b0d-fbf1-453f-b651-1a627b8d0d69

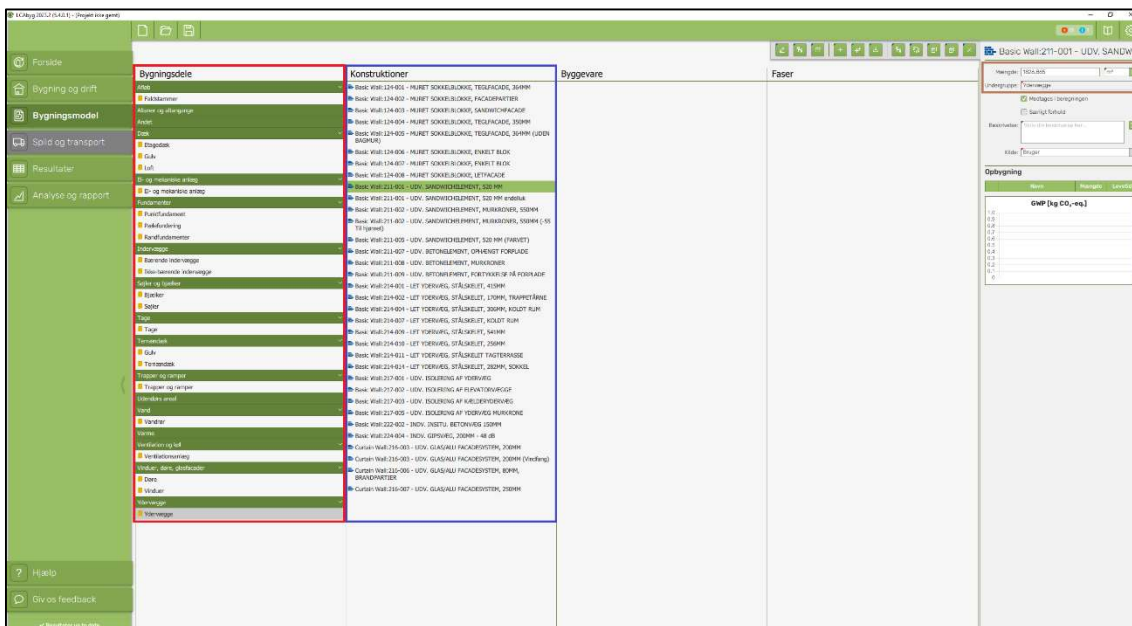
Figur 17. Del 1: Eksempel på LCI oprettet med scriptet.

Parameteret "Bygningsdel" anvendes til at strukturere dataet i LCAByg. Parametrene "Lifespan" og "Delayed start" bruges først, når scriptet "Update from LCAByg" eksekveres. De fem nye id-parametre benyttes i JSON-filerne, der oprettes for at formidle dataet i LCI'en til LCAByg. Efter data er blevet tilføjet til alle elementer i LCI'en eksporteres den til projektfolderen i Excel- og CSV-format.

Other_id	Element_id	Edge_id	LCA category	Construction_edge_id
353f8b717-2e6a-44d7-acc0-f50e69870001	0d9305042-427a-4718-a426-7ae73ca92155	6adca2b-e809-4ba4-bada-73fca2c2ff6b	b2f5244c-87e6-40d0-99f7-17b3d85a52a	9352161c-750b-4318-b4ed-e0f53c3a99f9
7002d72c-492d-4276-9f8a-7db299521402	6de1ba2c-0f71-4f64-8343-a42758c76274	66be28f1-56e6-4692-904a-153ce0a13ae3	9ab6709-61d1-46c1-4a2c-6d0b0dd0b9b1	7002d72c-492d-4276-9f8a-7db299521402
7002736cf-ae68-4ade-435b-bda752bb3ab9	33c1bc9-0191-4464-3074-c1b787677e98	7ad25f3c-a686-44da-8ed5-00c72123514a	40e0-00c9-9c1d-43ac-8e4c-6d4e04b48c5b	251155d4-ea41-439a-99fa-4ec1c870-2018
dea7121-334-4334-874-c73be03d630f	0dd1f318-5f55-4185-495a-10ca29aacef9	47edea83-83b6-4ba0-bae0-54445f3d9505	506a42e0-30c4-4849-4759-41f09b307037	16eb58bc-7978-4053-85e5-07372d3a604d
972221-411-451-8442-c43e03a0a36c3	12225f0-805e-4d8c-903a-792ac68cf3	bc26d764-437d-4ee9-a40b-76f335d54062	07347124-d27a-4242-936f-98f864c1d909	972221-411-451-8442-c43e03a0a36c3
5826747-1138-4343-9000-9366c18c187	001e01ff-c61c-4b9a-9133-0a2b6e8af3c0	ad5c6ee-af40-43d3-979b-795b0b000050	6ad1c4e05-8338-433d-979b-795b0b000050	5826747-1138-4343-9000-9366c18c187
308c24b4-7473-4284-99ff-a6183bb281a3	52c9579e-5f2a-4719-4713-3a47c3d16169	9106089-7863-4d7c-ad9f-35f050a1f58d	dc05551b-3189-44d2-902b-27e7c075b630	308c24b4-7473-4284-99ff-a6183bb281a3
75822498-4571-46be-88c3-31666ef9931		3121f1b6-9883-40d7-91d2-e0f9d0baf540	10a1b22d-4487-466a-96e2-d46351a10760	62d2b0d-8313-44cf-aeef-25a570c7020
				5947700-1b7f-453f-4651-1a627f8b0d69

Figur 18. Del 2: Eksempel på LCI oprettet med scriptet.

I scriptets sidste del udarbejdes 10 JSON-filer, som LCAbyg kræver for at åbne et projekt igennem JSON-integrationen. JSON-filerne oprettes ikke fra ny, da filerne på forhånd er importeret ind i softwareudviklingsmiljøet som skabeloner. Fremfor at opbygge JSON-filerne fra bunden udskiftes UUID'et i skabelonerne, så LCAbyg betragter JSON-filerne som et nyt projekt. Det sidste der sker i scriptet er, at JSON-filerne bliver eksporteret til projektfolderen, som efterfølgende kan åbnes i LCAbyg. I figur 19 er dataet fra LCI'en visualiseret igennem LCAbyg's brugergrænseflade, hvor den røde markering indeholder de 21 bygningsdele, der bliver brugt til at strukturere data. I den blå markering ses alle objekttypenavnene fra den valgte bygningsdel (yder-vægge). Den brune markering (oppe i højre hjørne) viser objekttypernes samlede mængde, enheden og LCAbyg-kategorien.



Figur 19. Dataet fra LCI'en visualiseret igennem LCBygs brugergrænseflade.

5.2 Update from LCAByg

Scriptet bruges til at opdatere LCI'en, når data er blevet tilføjet i LCAByg. Inden scriptet eksekveres skal brugeren eksportere projektet tilbage til projektfolderen fra LCAByg. I scriptet åbnes JSON-filerne fra projektfolderen, hvori det kontrolleres, om nye konstruktioner eller byggevarer er tilføjet. Hvis det er tilfældet, tilføjes de til den eksisterende LCI. Til sidst i scriptet eksporteres den nye LCI til projektfolderen, hvilket overskriver den eksisterende LCI.

Detaljeret gennemgang

I første del af scriptet importeres udvidelsespakkerne pandas og json, og stien til projektfolderen indsættes. Fra projektfolderen importeres tre JSON-filer (Construction, Element og Others), hvor alle konstruktioner og byggevarer i LCAByg-projektet opbevares. I softwaremiljøet er LCABygs database, indeholdt konstruktioner og byggevarer med deres navngivning og id, importeret ind som CSV-filer. For at identificere navnene på de konstruktioner eller byggevarer, der er blevet tilføjet i LCAByg, så sammenlignes id fra elementerne i JSON-filerne med de id som eksisterer i LCABygs databaser. Det er nødvendigt at gøre, fordi navnene på byggevarer og konstruktioner ikke eksisterer i JSON-filerne. De resterende informationer er dog tilgængelige i JSON-filerne.

Når alle nye byggevarer og konstruktioner er identificeret, bliver de tilføjet til den eksisterende LCI, der tilgås fra projektfolderen. Det er valgt, at måden der skelnes mellem byggevarer og konstruktioner i LCI'en på er ved at udfylde parameteret "Bygningsdel" med "x" og deres LCAByg-kategori med "Materiale". Ligeledes organiserer scriptet det således, at alle byggevarer indsættes under den konstruktion, som de er tilknyttet (se figur 20). Afslutningsvis eksporteres den opdaterede LCI tilbage til projektfolderen, hvorved den tidligere LCI overskrives.

Bygningsdel	LCA_category	Type_name	Amount	Unit	Lifespan	Delayed_start
Ydervægge	Ydervægge	Basic Wall:211-001 - UDV. SANDWIC	1826,865	M2	-	-
x	Materiale	Overflade, Facademaling, silikat	0,3	KG	10	0
x	Materiale	Beton vægelementer, 15 cm tyk væ	1	M2	120	0
x	Materiale	Mineraluld, alm.	0,3	M3	80	0
x	Materiale	Beton vægelementer, 20 cm tyk væ	1	M2	120	0
Ydervægge	Ydervægge	Basic Wall:211-001 - UDV. SANDWIC	8,464	M2	-	-

Figur 20. Udsnit af en opdateret LCI med fire byggevarer tilknyttet en ydervæg.

5.3 Update from IFC

Når en ny version af bygningsmodellen eksporteres til IFC, tages dette script i brug. Scriptet fungerer på måde som det første script, indtil LCI'en skal opdateres. Forinden opdateringen sammenlignes IFC-modellerne, så data fra eksisterende bygningsobjekter ikke overskrives. Bygningsobjekternes GUID bruges til at identificere eksisterende og nye objekter. Alle ændringer i bygningsmodellen noteres i en separat Excel- og CSV-fil, så brugeren kan danne sig et overblik. Dernæst oprettes den nye LCI, hvor alt data tilført i LCAByg indhentes fra den eksisterende LCI. Til slut eksporteres LCI'en og JSON-filerne opdateres, så det opdaterede data kan åbnes i LCAByg.

Detaljeret gennemgang

Scriptets indledende del er allerede beskrevet i afsnittet "Create from IFC", så gennemgangen startes en tredjedel inde i scriptet, når der kontrolleres for eksisterende og nye bygningsobjekter i den opdaterede IFC-model. Kontrollen foregår ved at åbne Excel-filen fra projektfolderen, der indeholder alle IFC-objekter fra den tidligere IFC-model, og udtrække alle objekters GUID. Hvis et objekt fra den nye IFC-model har samme GUID, så ved scriptet, at objektet eksisterer i LCI'en. Alle eksisterende objekter sorteres fra nye objekter, da nye objekter kan overføres til en ny LCI uden komplikationer. Det kræver derimod en yderligere gennemgang, før eksisterende objekter kan overføres, da deres mængde eller navngivning kan være ændret. Scriptet sammenligner derfor objekternes data fra den tidligere IFC-model med data fra den nye IFC-model. Såfremt der er forskel på dataet, anvendes det nye data, når det skal overføres til LCI'en. Scriptet opretter dog en ny Excel- og CSV-fil som dokumenterer alle ændringer, så brugeren får et indblik i, hvad der er sket med bygningsmodellen. Et eksempel herpå kan ses i figur 21, hvor 15 ændringer er noteret.

	Type_name		Amount	
	Old	New	Old	New
0			0,026	0,044
1			39,995	35,695
2	M_Fixed:0915 x 1830mm	M_Fixed:0915 x25003mm	1,674	2,287
3			14,022	20,805
4	M_Fixed:0915 x 1830mm	M_Fixed:0915 x25003mm	1,674	2,287
5			59,635	40,684
6	M_Single-Flush:0915 x 2134mm	M_Single-Flush:Office Door		
7	M_Fixed:0915 x 1830mm	M_Fixed:0915 x25003mm	1,674	2,287
8			11,475	19,532
9			22,243	10,835
10			15,126	20,219
11			23,447	30,293
12	M_Fixed:0915 x 1830mm	M_Fixed:0915 x25003mm	1,674	2,287
13	M_Fixed:0915 x 1830mm	M_Fixed:0915 x25003mm	1,674	2,287
14			16,356	27,462

Figur 21. Udsnit fra Excel-fil der dokumenterer ændringer i bygningsmodellen.

Hvis objekter er ændret, men stadig bibeholdt i IFC-modellen, får objektet et løbenummer, så brugeren kan tracke antallet af ændringer. I samme Excel- eller CSV-fil nedskrives også, hvilke objekter er tilføjet i IFC-modellen og hvilke er fjernet fra modellen. Det vises under løbenumrene med teksten "DELETED" eller "ADDED" samtidig med, at der i kolonnerne ved siden af er tilføjet data om objekterne, så brugeren har nemmere ved at lokalisere objekterne (figur 22). Samme hjælpende datakolonner er tilføjet ved objekterne med løbenumre, selvom det ikke kan ses i figur 21.

	IFC_entity_new	LCA_category_new	Type_name_new	Amount_new	Unit_new	GUID_new	LCA_category_id_new
DELETED	IfcBeam	Bjælker	UB-Universal Beams:UB305x165x40	0,014	M3	00p2HhMaz5SuW\$JEIk5jqP	ec9ee040-9c1d-4cae-864c-4c6a0e4b8c5b
DELETED	IfcWallStandardCase	Ikke-bærende indervægge	Basic Wall:Generic - 90mm Brick	7,653	M2	0uXlp\$4ZL7gen4kpNT6bd8	a9ab6709-61e1-46c1-a2e8-df00b0dd0bb91
DELETED	IfcWindow	Vinduer	M_Fixed:0915 x 1830mm	1,674	M2	2HAAFR_OH52B9xUJEj\$piB	dcfb55db-3189-4d22-9b2b-2e27c057b363
DELETED	IfcWindow	Vinduer	M_Fixed:0915 x 1830mm	1,674	M2	2HAAFR_OH52B9xUJEj\$piB	dcfb55db-3189-4d22-9b2b-2e27c057b363
ADDED	IfcBeam	Bjælker	Beam_main:UB305x165x40	0,05	M3	0i37znfmP03wnBxYkqJV1y	ec9ee040-9c1d-4cae-864c-4c6a0e4b8c5b
ADDED	IfcBeam	Bjælker	Bjælke:UB305x165x40	0,009	M3	0i37znfmP03wnBxYkqJOv9	ec9ee040-9c1d-4cae-864c-4c6a0e4b8c5b
ADDED	IfcBeam	Bjælker	Bjælke:UB305x165x40	0,014	M3	0i37znfmP03wnBxYkqJOvU	ec9ee040-9c1d-4cae-864c-4c6a0e4b8c5b
ADDED	IfcDoor	Døre	Dør:0915 x 2134mm	1	Pcs	0i37znfmP03wnBxYkqJOoJ	9bd14e05-5fba-4ecb-ab41-7940a8a0dbdc
ADDED	IfcDoor	Døre	Dør:0915 x 2134mm	1	Pcs	0i37znfmP03wnBxYkqJOm_	9bd14e05-5fba-4ecb-ab41-7940a8a0dbdc
ADDED	IfcDoor	Døre	M_Single-Flush:Office Door	1	Pcs	0i37znfmP03wnBxYkqJOqq	9bd14e05-5fba-4ecb-ab41-7940a8a0dbdc
ADDED	IfcSlab	Gulv	Floor:Standard Timber-Wood Finish	16,642	M2	0i37znfmP03wnBxYkqJOeA	bf2534e2-87e6-40cb-9fff-17fb3d85a52a
ADDED	IfcWallStandardCase	Ikke-bærende indervægge	Basic Wall:Generic - 90mm Brick	2,841	M2	0i37znfmP03wnBxYkqJOx6	a9ab6709-61e1-46c1-a2e8-df00b0dd0bb91
ADDED	IfcWallStandardCase	Ikke-bærende indervægge	Basic Wall:Generic - 90mm Brick	5,7	M2	0i37znfmP03wnBxYkqJOtz	a9ab6709-61e1-46c1-a2e8-df00b0dd0bb91
ADDED	IfcColumn	Søjler	UC-Universal Columns-Column:Column	0,032	M3	0i37znfmP03wnBxYkqJOuc	506a420e-c3cd-4849-a759-3117f10937b1
ADDED	IfcColumn	Søjler	UC-Universal Columns-Column:Column	0,032	M3	0i37znfmP03wnBxYkqJOIY	506a420e-c3cd-4849-a759-3117f10937b1

Figur 22. Udsnit fra Excel-fil der dokumenterer ændringer i bygningsmodellen.

Excel- og CSV-filen eksporteres herefter til projektfolderen sammen med det nye udtræk fra bygningsmodellen. Sideløbende oprettes en ny folder i projektfolderen som gemmer udtrækkene fra den gamle IFC-model. Hernæst oprettes den nye LCI med opdaterede data. Inden dataet kan overføres til LCI'en, tjekkes den eksisterende LCI igennem, så det sikres at LCA-data tilknyttet objekterne også overføres. Endvidere er det nødvendigt at kontrollere for, om objekter manuelt er blevet tilføjet til LCI'en (uddybes i script fem), da disse objekter ikke fremgår i udtrækket fra IFC-modellen. Scriptet afsluttes med at eksportere den nye LCI og opdatere JSON-filerne i projektfolderen, så LCI'en kan åbnes i LCAByg.

5.4 Create from LCI

Scriptet kan anvendes af brugeren, hvis der ikke ønskes at oprette et LCAByg-projekt ud fra en IFC-model. Størstedelen af scriptet er tidligere beskrevet, da fremgangsmåden er den samme som ved første script. Den eneste forskel er, at der anvendes en tom LCI, hvori brugeren manuelt skal indtastet dataet, som i første script blev udtrukket fra en IFC-model. For hvert objekt skal brugeren indtaste bygningsdel, LCA-kategori, typenavn, mængde og enhed. Når scriptet eksekveres, tilføjes data til de tomme felter i LCI'en, så resten af scriptet kan udføres på samme facon som første script. Da funktionerne i scriptet allerede er beskrevet, indeholder dette afsnit ikke nogen detaljeret gennemgang af scriptet.

5.5 Update from LCI

Det sidste script kan benyttes i situationer, hvor brugeren skal inddrage objekter i et LCAByg-projekt som ikke figurerer i IFC-modellen. Scriptet fungerer på samme måde som "Create from LCI" scriptet, men uden nødvendigheden for at oprette de 10 JSON-filer, som påkræves for at åbne et LCAByg-projekt, da filerne i forvejen eksisterer. De manuelle indtastede objekter tilføjes blot til de JSON-filer, der indeholder de eksisterende bygningsobjekter (Construction, Element og Others). Scriptet opdaterer eksisterende filer og kan derfor først anvendes, når et LCAByg-projekt allerede er oprettet. Ingen nye funktioner findes i scriptet, dermed udebliver den detaljerede gennemgang.

6. Feedback og videreudvikling

Prototypen blev præsenteret hos brugeren, hvor en LCA-specialist, en bygningskonstruktør og to BIM-specialister var til stede. Præsentationen havde en varighed på cirka 15 minutter efterfulgt af 45 minutters feedback og diskussion. I dette afsnit gengives feedbacken på prototypen samt idéerne til prototypens videreudvikling.

Grundlæggende blev prototypen anset som et værdiskabende værktøj, der kunne eliminere mange af de eksisterende manuelle processer. Det blev eksemplificeret med en gængs situation fra praksis, hvor tidlige LCA-screeninger skulle udføres. LCA-specialisten ville nogle gange starte sit arbejde i LCAByg, fremfor at oprette en LCI i Excel. Hvis arbejdet blev startet i LCAByg, så var LCA-specialisten nødsaget til manuelt at genindtaste data i LCI'en, fordi der ikke var tovejskommunikation mellem LCAByg og LCI'en. Prototypen kan kommunikere begge veje, og dermed kan genindtastninger og manuelle opdateringer af LCI'en undgås.

Forskellige funktioner blev efterspurgt i prototypen, hvoraf muligheden for at modificere og kommentere på data i LCI'en blev vægtet højt. Det blev belyst, at mængder fra bestemte bygningsobjekttyper ofte bliver ganget med en faktor, så et mere retvisende billede af mængdeforbruget i praksis fås. Yderligere blev det fortalt, at objekter kunne risikere at blive opdelt, såfremt de skulle indgå i forskellige LCA-beregninger. Det skal derfor være muligt at ændre på objektmængder samtidig med, at det tydeligt skal fremgå, hvilke objekter er blevet modificerede. I prototypens LCI er det muligt manuelt at ændre bygningsobjekters mængder, men det vil ikke fremgå nogen steder. En ekstra kolonne kan dog uden problemer tilføjes til prototypens LCI, så brugeren har mulighed for at tilknytte kommentarer til manuelle ændringer. En kolonnetilføjelse giver derfor god mening, men løser ikke problemet, da manuelle mængdeændringer vil blive overskredet, når bygningsmodellen opdateres. Af den grund bør det være muligt at fastlåse mængder for bygningsobjekter i LCI'en, hvoraf det tydeligt skal fremgå, hvilke mængder er blevet fastlåst.

I prototypen kan konstruktioner manuelt tilføjes til LCI'en, således at objekter, som ikke eksisterer i IFC-modellen, kan inddrages i LCA-beregningen. Det er ikke muligt at tilføje byggevarer i prototypen, da byggevarer tilføjjelsen var tiltænkt at skulle ske i LCAByg. Det blev dog påpeget, at det på store projekter, hvor mange konstruktionsopbygninger næsten er identiske, ville være smart, hvis byggevarer kunne kopieres i LCI'en. Derved kunne det undgås, at den samme byggevarer skulle tildeles samtlige konstruktioner i LCAByg. Det er et tidsbesparende forslag, da det er hurtigere at kopiere byggevarer i LCI'en fremfor at tilføje dem i LCAByg.

Til at øge gennemsigtigheden, af hvilke IFC-objekter prototypen inddrager fra bygningsmodellen, blev en BIM-viewer foreslået. Brugeren har et behov for at kunne gennemskue, hvad der sker undervejs i LCA-pro-

cessen. På nuværende tidspunkt er prototypen ikke i stand til at håndtere alle undertyper af IfcBuildingElement, hvormed en masse objekter fra bygningsmodellen mangler i LCAbyg. En BIM-viewer, der kan forbinde bygningsmodellen med prototypen, vil gøre det lettere for brugeren at navigere i, hvilke bygningsobjekter mangler i LCAbyg. I prototypen er det muligt at identificere, hvilke bygningsobjekter er indeholdt, da objekternes GUID udtrækkes. Det er dog ikke muligt at se, hvilke bygningsobjekter mangler at blive udtrukket, hvilket en BIM-viewer opkoblet til bygningsmodellen ville kunne visualisere.

Der blev stillede spørgsmål til, hvorvidt det var hensigtsmæssigt at udtrække mængder fra blot ét filformat. Det blev diskuteret, om IFC-formatet var omfattende nok på grund af de manglende Base Quantities for bestemte objekttyper. Udfordringen, med at vægobjekter klassificerede som IfcWall blev eksporteret uden Base Quantities, vakte bekymring, da det er uvist, hvornår der skelnes mellem IfcWall og IfcWallStandardCase. At videreudvikle prototypen til at fungere med Navisworks (softwareprogram fra Autodesk til 3D-modelgennemgang) blev foreslået, da programmet kan håndtere mange forskellige filformater. Det vil give brugeren flere alternativer til mængdeudtræk fra bygningsmodeller, hvilket potentielt kan overkomme manglerne i IFC-skemaet. Modsat vil det stille krav til, at brugeren har adgang til Navisworks (lukket BIM). Som et modsvar til Navisworks-forslaget blev det pointeret, at udfordringen ikke var IFC-skemaet men derimod dårlig modelleringsskik. Med det menes, at de manglende Base Quantities for IfcWall-objekter ikke er en udfordring, såfremt de projekterende modellerer korrekt.

Blandt de tilstedeværende personer var der enighed om, at prototypens nuværende funktioner kunne forbedre deres workflow. Prototypen blev dog ikke set som en erstatning til deres nuværende værktøjer men derimod som en tilføjelse. En stor værdi blev set i at anvende prototypen som et mellemlid mellem deres nuværende LCI-skabelon og LCAbyg. Derved fås tovejskommunikation mellem prototypen og LCAbyg samtidig med, at deres LCI kan modificeres i et frakoblet dokument. Ved den metodik minimeres sandsynligheden for, at brugeren ved et uheld laver uønsket ændringer. Der er ikke nogen tilbageknap i prototypen. Hvis brugeren opdaterer data, så kan det blive udfordrende at vende tilbage til det tidligere data. Det skaber en sikkerhed for brugeren at have et frakoblet dokument, hvor processen styres fra.

Det blev påtalt, at brugeren ikke ville have kendskab til den bagvedliggende programmering i prototypen. Dens funktioner skal derfor køre automatisk i baggrunden, så brugeren kun skal forholde sig til outputtet. En brugergrænseflade vil dermed skabe stor værdi for brugeren i en videreudvikling.

7. Diskussion

Diskussionsafsnittet er inddelt i to underkapitler, hvor det første relaterer sig til specialets metodik, der sammenlignes med fremgangsmåden fra Contextual Design. Det sidste omhandler prototypen, dens udviklingsforløb og videreudvikling. Der reflekteres over beslutningerne taget undervejs, og feedbacken fra brugeren inddrages og diskuteres.

7.1 Metode

Det kan diskuteres, hvorvidt specialets metodik manglede en yderligere aktivitet inden udviklingen af prototypen blev påbegyndt. I et Contextual Design forløb efter Holtzblatt og Beyers (2017) retningslinjer, anbefaler de, at det grundigt planlægges, hvilke funktioner og elementer produktet (prototypen) skal indeholde. Holtzblatt og Beyer advokerer for at benytte storyboards og User Environment Design til planlægningsprocessen, da metoderne fungerer som effektive analyseværktøjer til klarlægningen af nødvendige funktioner og elementer. Den anvendte metodik var mere iterativ anlagt, hvor kontakten med brugeren blev brugt til at planlægge prototypens funktioner. Prototypens succes afhænger derfor af en god sparring med og konstruktiv feedback fra brugeren, der bliver medudviklet på prototypen. Hvis det ønskes at være mindre afhængig af brugerens feedback, anses storyboards og User Environment Design som brugbare metoder at benytte.

Holtzblatt og Beyer angriber ligeledes udviklingsforløbet af produktet anderledes, da de foreslår først at udarbejde en papirprototype til at validere produktdesignet, inden ressourcer bruges på produktudviklingen. Anvendelsen af papirprototypen begrundes ud fra hjemmelen, jo tidligere et problem opdages i en udviklingsproces, desto nemmere og billigere er det at rette op på problemet. Med andre ord vil det være mere besværligt og kosteligt at lave ændringer til prototypen efter, at ressourcer er forbrugt på produktudviklingen. På store og komplekse projekter kan der potentielt spares mange udviklingsressourcer, hvis fejl og mangler identificeres på papirprototypen (Holtzblatt & Beyer, 2017). Det er dog en tidskrævende proces at udarbejde en papirprototype og teste den med brugeren, hvilket er årsagen til, at det blev fravalgt. I en videreudvikling af prototypen, når en brugergrænseflade skal designes, bør det genovervejes, om papirprototypen skal ibrugtages. Det kan være meget tidskrævende at lave designændringer, så grundig designplanlægning på en papirprototype vil potentielt kunne spare mange ressourcer.

7.2 Prototypen

Til trods for de adskillige funktioner som blev inkorporeret i prototypen efter konceptdemonstrationen, er der stadig mange områder, hvorpå prototypen kan optimeres. En brugergrænseflade og online adgang til prototypen vil i særdeleshed være optimalt at arbejde på. I prototypens nuværende version kræver det et stort kendskab til den bagvedliggende programmering, hvis en bruger skal anvende den. En simpel brugergrænseflade der kan give feedback til brugeren, hvis fejl opstår, vil øge brugervenligheden af prototypen. Det kunne eksempelvis være noget så enkelt som at tilføje knapper, så brugeren kunne vælge sine filer på computeren fremfor at skulle indtaste den korrekte sti direkte i PyCharm. Et andet eksempel kunne være en advarsel som poppede op, hvis brugeren manglede at indtaste noget data. Forinden detaljer som dette udvikles, kræver det, at prototypen ikke opsættes til en specifik computer, som det er tilfældet nu. I en videreudvikling bør det derfor undersøges, hvordan prototypen bedst kan publiceres, så den bliver let tilgængelig.

Idéen, med at opbevare data online så det også kan bruges i andre henseender, bør ligeledes være et fokusområde i videreudviklingen. I det lange løb handler det om at få strømlinede sine data, så data kan genbruges til forskellige formål fremfor at skulle udtrækkes til hvert formål. Det er selvfølgelig ikke garanteret, at alle dataudtræk kan anvendes universelt, men et mængdeudtræk fra bygningsobjekter vil uden tvivl kunne anvendes til adskillige formål. Det vil dog komplicere programmeringen markant, da det vil kræve, at mindst ét Web API bliver indkodet i prototypen. I den nuværende version kan prototypens fem scripts køres lokalt på computeren, hvilket af den årsag simplificerer programmeringen.

En endnu større programmeringsændring påkræves, hvis en BIM-viewer skal tilkobles prototypen, så det kan kontrolleres, hvilke bygningsobjekter fra IFC-modellen mangler at udtrækkes. Det kan diskuteres, hvorvidt en BIM-viewer overhovedet er den optimale kontrolproces. Der kan argumenteres for, at tilkoblingen af en BIM-viewer er at overkomplicere en kontrolproces, der vil kunne udføres på en noget enklere måde. Eksempelvis kunne alle bygningsobjekter tilføjes et nyt parameter eller en værdi i et eksisterende parameter, som der i BIM-softwaret kunne filtreres efter. Det kunne også gøres ved at prototypen talte antallet af bygningsobjekter, der blev udtrukket, så antallet af objekter kunne sammenstilles. Muligheden for at tilgå og visualisere sit data i 3D vil klart øge gennemsigtigheden for brugeren, men spørgsmålet er, om det er en funktion, som prototypen skal tilvejebringe.

Det vigtigste i en videreudvikling er dog at få udvidet antallet af bygningsobjekttyper, som prototypen kan håndtere. På brugerens bygningsmodel var det tydeligt, at mange bygningsobjekter manglede i udtrækket. Prototypen skal ikke nødvendigvis kunne håndtere alle undertyper af `IfcBuildingElement`, da f.eks. IFC-objekter klassificerede som `IfcBuildingElementProxy` vil være ekstremt svære, hvis ikke umulige, at standardisere et mængdeudtræk for. `IfcBuildingElementProxy`-objekter bruges som stedfortræder, når bygningsobjekter

ikke kan klassificeres andre undertyper af IfcBuildingElement. Det kan dermed ikke garanteres, at bestemte data er tilgængelige, hvilket forårsager standardiseringsudfordringen. Dog mangler prototypen at kunne håndtere essentielle bygningsobjekter som lofter (IfcCovering), fundamenter (IfcFooting), trapper (IfcStairFlight), rækværker (IfcRailing) og lignende objekttyper, der ofte eksisterer i bygningsmodeller. Det kan undersøges, hvilke mængdedata kan udtrækkes fra objekttyperne, så det vides, om deres inddragelse i prototypen vil skabe udfordringer eller ej. Informationen er tilgængelig i buildingSMARTs officielle dokumentation for IFC2X3 TC1-skemaet. For IfcCovering kan brutto- og nettoarealet udtrækkes, men IfcFooting, IfcStairFlight og IfcRailing har ingen standard mængdeparametre tilknyttet. Det er dog ikke ens betydende med, at IFC-objekttyperne ikke kan have mængder tilknyttet dem. I brugerens bygningsmodel var kun bruttoarealet tilgængeligt for IfcCovering-objekter, hvorimod IfcStairFlight-objekters længde og nettovolumen kunne udtrækkes. Ingen mængder kunne udtrækkes fra IfcRailing-objekter, og ingen IfcFooting-objekter eksisterede i bygningsmodellen, da IfcWallStandardCase blev brugt til linjefundamenterne. For videreudvikling vil det betyde, at prototypen støder på store udfordringer, når flere objekttyper skal inddrages.

Det er igennem prototypeudviklingen belyst, at IFC-skemaet har nogle klare begrænsninger. Det er derfor nødvendigt at reflektere over, om mængdeudtrækkene fra bygningsmodellen skal være mulige fra andre filformater end IFC. Til præsentation og feedback-sessionen hos brugeren blev det foreslået at udtrække mængder fra Navisworks, da programmet kan håndtere mange filformater. Det kunne potentielt godt være en løsning til at overkomme IFC-skemaets begrænsninger, men det vil være en falliterklæring for openBIM. I stedet bør der fokuseres på, hvordan IFC-skemaet kan udvides, så mængdeudtrækkene bliver gjort mulige. Det kan eksempelvis gøres ved at modificere indstillingerne i en IFC-eksport, så bygningsobjekter bliver eksporterede med deres tilhørende mængdeskemaer som et IFC-egenskabssæt. Det vil dermed være muligt at standardisere, hvordan prototypen skal udtrække data fra det eksporterede IFC-egenskabssættet. Slagsiden ved denne fremgangsmåde er, at der skal bruges tid på at opsætte mængdeskemaer i BIM-softwaret. Ligeledes kan der argumenteres for, at IFC-eksporten bliver overflødig, da mængdeskemaerne i stedet kan eksporteres til Excel, hvorfra data hurtigt kan overføres til LCI'en. Første aktivitet i prototypens videreudvikling bør derfor være at identificere og standardisere en metode for, hvordan mængder udtrækkes fra bygningsobjekter i IFC-modeller.

8. Konklusion

I forsøget på at reducere de store mængder energi og ressourcer der forbruges på opførelse, drift og vedligehold af bygninger og bygværker blev et klimapåvirkningskrav indført i det danske bygningsreglement. Klimapåvirkningerne skal udregnes ved at udføre LCA-beregninger med data fra bygningsmodeller. Adskillige udfordringer er dog blevet identificerede i dataudvekslingen mellem bygningsmodeller og LCA-beregningsværktøjer. Det blev derfor undersøgt, hvordan IFC kan understøtte udarbejdelsen af LCA-beregninger og forbedre dataudvekslingen.

En brugercentreret metode, der tager udgangspunkt i grundelementerne fra Contextual Design, blev udarbejdet, og en stor rådgivende arkitekt- og ingeniørvirksomheds LCA-processer blev undersøgt. Yderligere blev et systematisk litteraturreview med snowballing som søgeteknik udført, hvilket identificerede LCA relaterede udfordringer med IFC-skemaet samt udfordringer med BIM-LCA integrationen. De identificerede udfordringer blev sammenstillet med undersøgelsen af virksomhedens LCA-processer, så en afgrænsning af udfordringerne kunne udføres. Herefter blev udarbejdelsen af en prototype påbegyndt, hvor det blev valgt at anvende programmeringssproget Python og programmere i et integreret softwareudviklingsmiljø kaldet PyCharm. Prototypen blev udviklet for at optimere virksomhedens eksisterende LCA-workflow, hvilket blev gjort ved at udtrække data fra IFC-modeller og efterfølgende integrere det i LCA-beregningsværktøjet LCAbyg.

Igennem feedback og dialog med virksomheden blev flere funktioner identificerede og indarbejdede i prototypen. Det er derfor muligt at udtrække data fra IFC-modeller og integrere det i LCAbyg, men også at formidle data fra LCAbyg tilbage til prototypen. Endvidere kan data i prototypen opdateres uden at udtrække dem fra IFC-modeller, da det kan være nødvendigt at indhente data fra andre steder end bygningsmodellen. Da prototypen afslutningsvist blev præsenteret hos virksomheden, blev den betragtet som et værdiskabende værktøj, der kunne eliminere mange manuelle processer i deres eksisterende workflow. Prototypen er dog langt fra at være et endeligt produkt, da den i sin nuværende version kan håndtere følgende undertyper af IfcBuildingElement: IfcBeam, IfcColumn, IfcCurtainWall, IfcDoor, IfcRoof, IfcSlab, IfcWall og IfcWindow. Det er dermed ikke alle bygningsobjekttyper der kan håndteres, men i en videreudvikling kan flere tilføjes.

Flere store barrierer i BIM-LCA integrationen blev identificerede undervejs i prototypeudviklingen, hvoraf de fleste relaterede sig til mængdeudtrækket fra IFC-modellen, da flere bygningsobjekter blev eksporterede uden mængder. Det belyste en udfordring ved IFC-skemaet, da objekternes mængder var tilgængelige i BIM-softwaret. Udfordringen er, blandt de håndterbare IFC-objekttyper i prototypen, gældende for glasfacadesystemer og bestemte vægge. Blandt prototypens manglende IFC-objekttyper er udfordringen mere udbredt. I fremtidige studier er det derfor oplagt at undersøge, hvordan IFC-skemaet kan udvides, så en standardiseret

metode for mængdeudtræk af IFC-objekter kan udarbejdes. Ligeledes bør det undersøges, hvad der forårsager, at vægobjekter eksporteres som IfcWall fremfor IfcWallStandardCase.

I en videreudvikling af prototypen kan en brugergrænseflade udvikles, så prototypens anvendelse bliver mere brugervenlig. En anden brugbar tilføjelse vil være at udvikle en funktion som kan kontrollere, om alle bygningsobjekter fra bygningsmodellen eksisterer i prototypen. At integrere Web API'er, så prototypen kan indhente og uploade data online, vil også være anvendeligt.

9. Litteraturliste

Alwan, Z. & Ilhan Jones, B., 2022. IFC-based embodied carbon benchmarking for early design analysis. *Automation in Construction*. 142(104505), s. 1-12. doi: 10.1016/j.autcon.2022.104505.

Andriamamonjy, A., Saelens, D. & Klein R., 2018. An automated IFC-based workflow for building energy performance simulation with Modelica. *Automation in Construction*. 91(1), s. 166-181. doi: 10.1016/j.autcon.2018.03.019.

Booth, A., Papaioannou, D. & Sutton, A., 2016. *Systematic Approaches to a Successful Literature Review*. 2. udgave. Los Angeles: Sage, s. 108-140.

Bruhn, S. & Sørensen, C. G., 2020. *Json guide for LCAbyg version 5.0*. BUILD - Institut for Byggeri, By og Miljø, Aalborg Universitet, A.C. Meyers Vænge 15, 2450 København SV. [Lokaliseret d. 06.01.2024]. Tilgængelig fra: <https://www.lcabyg.dk/da/bim-integration-da/>.

BuildingSMART, 2007. IFC2X3 TC1 (ISO/PAS 16739:2005). [Online]. [Lokaliseret d. 10.01.2024]. Tilgængelig fra: <https://standards.buildingsmart.org/IFC/RELEASE/IFC2x3/TC1/HTML/>.

BuildingSMART, 2017. IFC4 ADD2 TC1 (ISO 16739-1:2018). [Online]. [Lokaliseret d. 10.01.2024]. Tilgængelig fra: https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/.

Bygningsreglementet 2018. (BEK nr 1464 af 28/11/2022). [Online]. København: Bolig- og Planstyrelsen. [Lokaliseret d. 03.01.2024]. Tilgængelig fra: <https://www.retsinformation.dk/eli/lta/2022/1464>.

Dansk standard, 2008. Miljøledelse – Livscyklusvurdering – Principper og struktur. DS/EN ISO 14040. 4. udgave. [Lokaliseret d. 24.12.2023].

Dansk standard, 2017. Bygningsinformationsmodeller – Manual for informationsleverance (IDM) – Del 1: Metodik og format. DS/EN ISO 29481-1:2017. 1. udgave. [Lokaliseret d. 26.12.2023].

Dansk standard, 2019. Bæredygtighed inden for byggeri og anlæg – Miljøvaredeklarationer – Grundlæggende regler for produktkategorien byggevarer. DS/EN 15804:2012+A2:2019. 1. udgave. [Lokaliseret d. 25.12.2023].

Deng, X. & Lu, K., 2023. Multi-level assessment for embodied carbon of buildings using multi-source industry foundation classes. *Journal of Building Engineering*. 72(106705), s. 1-21. doi: 10.1016/j.jobbe.2023.106705.

De Wolf, C., Cordella, M., Dodd, N., Byers, B. & Donatello, S., 2023. Whole life cycle environmental impact assessment of buildings: Developing software tool and database support for the EU framework Level(s). *Resources, Conservation and Recycling*. 188(106642). doi: 10.1016/j.resconrec.2022.106642.

Di Bari, R., Horn, R., Bruhn, S., Alaux, N., Saade, M., Soust-Verdaguer, B., Obrecht, T., Hollberg, A., Birgisdóttir, H., Passer, A. & Frischknecht, R., 2022. Buildings LCA and digitalization: Designers' toolbox based on a survey. *IOP Conf. Series: Earth and Environmental Science*. 1078(012092), s. 1-11.

doi: 10.1088/1755-1315/1078/1/012092.

DiKon, BIM7AA & Molio, 2022. Bygningsspecifikationer. 5. udgave. [Lokaliseret 26.12.2023].

Tilgængelig fra: <https://anvisninger.molio.dk/gratis-vaerktojer/bygningsspecifikationer/bygningsspecifikationer>.

Dominte, I., 2023. *Web API Development for the Absolute Beginner*. 1. udgave. Berkeley, CA: Apress, s. 3-18.

doi: 10.1007/978-1-4842-9348-5.

Eastman, C.M., Teicholz, P.M., Sacks, R. & Lee, G., 2018. *BIM handbook : A guide to building information modeling for owners, managers, designers, engineers and contractors*. 3. udgave. Hoboken, New Jersey: Wiley. s. 1-31. doi: 10.1002/9781119287568.

Figl, H., Ilg, M. & Battisti, K., 2019. 6D BIM—Terminal: Missing Link for the design of CO2neutral buildings. *IOP Conf. Series: Earth and Environmental Science*. 323(012104), s. 1-6.

doi: 10.1088/1755-1315/323/1/012104.

Forth, K., Abualdenien, J. & Borrmann, A., 2023. Calculation of embodied GHG emissions in early building design stages using BIM and NLP-based semantic model healing. *Energy and Buildings*.

284(112837), s. 1-17. doi: 10.1016/j.enbuild.2023.112837.

Guignone, G., Calmon, J.L., Vieira, D. & Bravo, A., 2023. BIM and LCA integration methodologies: A critical analysis and proposed guidelines. *Journal of Building Engineering*. 73(106780), s. 1-22.

doi: 10.1016/j.jobbe.2023.106780.

Holtzblatt, K. & Beyer, H., 2017. *Contextual Design*. 2. udgave. Amsterdam: Morgan Kaufmann.

doi: 10.1016/B978-0-12-800894-2.01001-2.

Ilhan, B. & Yaman, H., 2016. Green building assessment tool (GBAT) for integrated BIM-based design decisions. *Automation in Construction*. 70(1), s. 26-37. doi: 10.1016/j.autcon.2016.05.001.

Indenrigs- og Boligministeriet, 2021. National strategi for bæredygtigt byggeri. [Lokaliseret d. 26.12.2023].

ISBN: 978-87-971298-8-3. Tilgængelig fra: <https://im.dk/Media/637787884257325807/National%20strategi%20for%20b%C3%A6redygtigt%20byggeri-a.pdf>.

Jrade, A. & Abdulla, R., 2012. Integrating Building Information Modeling and Life Cycle Assessment Tools to Design Sustainable Buildings. *Proceedings of the CIB W78 2012: 29th International Conference –Beirut, Lebanon, 17-19 October*. [Online]. s. 1-10. [Lokaliseret 12.11.2023].

Tilgængelig fra: <https://itc.scix.net/pdfs/w78-2012-Paper-24.pdf>.

Klumbyte, E., Georgali, P.-Z., Spudys, P., Giama, E., Morkunaite, L., Pupeikis, D., Jurelionis, A. & Fokaides, P., 2023. Enhancing whole building life cycle assessment through building information modelling: Principles and best practices. *Energy and Buildings*. 296(113401), s. 1-12. doi: 10.1016/j.enbuild.2023.113401.

Kulahcioglu, T., Dang, J. & Toklu, C., 2012. A 3D analyzer for BIM-enabled Life Cycle Assessment of the whole process of construction. *HVAC & R Research*. 18(1-2), s. 283-293.
doi: 10.1080/10789669.2012.634264.

Lai, H., Zhou, C. & Deng, X., 2019. Exchange requirement-based delivery method of structural design information for collaborative design using industry foundation classes. *Journal of Civil Engineering and Management*. 25(6), s. 559-575. doi: 10.3846/jcem.2019.9870.

Level(s): Putting whole life carbon principles into practice, 2021. Luxembourg: Publications Office of the European Union. [Online]. [Lokaliseret 12.11.2023]. Tilgængelig fra: <https://op.europa.eu/en/publication-detail/-/publication/20761f2e-143f-11ec-b4fe-01aa75ed71a1/language-en/format-PDF/source-230076864>

LLatas, C., Soust-Verdaguer, B., Hollberg, A., Palumbo, E. & Quiñones, R., 2022. BIM-based LCSA application in early design stages using IFC. *Automation in Construction*. 138(104259), s. 1-14.
doi: 10.1016/j.autcon.2022.104259.

Loh, E., Dawood, N. & Dean, J., 2007. Integration of 3D Tool with Environmental Impact Assessment (3D EIA). *3rd Inter- Natl ASCAAD Conf. Embodying Virtual Archit., Alexandria, Egypt, 2007*. s. 51-66.

Ma, Z. & Zhao, Y., 2008. Model of Next Generation Energy-Efficient Design Software for Buildings. *Tsinghua Science & Technology*. 13(1), s. 298-304. doi: 10.1016/S1007-0214(08)70165-2.

Nadeem, A., Wong, A.K.D. & Wong, F.K.W., 2015. Bill of Quantities with 3D Views Using Building Information Modeling. *Arabian Journal for Science and Engineering*. 40(9), s. 2465–2477.
doi: 10.1007/s13369-015-1657-2.

Pauwels, P. & Terkaj, W., 2016. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*. 63(1), s. 100–133. doi: 10.1016/j.autcon.2015.12.003.

Pinheiro, S., Wimmer, R., O'Donnell, J., Muhic, S., Bazjanac, V., Maile, T., Frisch, J. & van Treeck, C., 2018. MVD based information exchange between BIM and building energy performance simulation. *Automation in Construction*. 90(1), s. 91-103. doi: 10.1016/j.autcon.2018.02.009.

Rasmussen, F.N., Trigaux, D., Alsema, E., Balouktsi, M., Birgisdóttir, H., Bohne, R., Dixit, M., Dowdell, D., Francart, N., Frischknecht, R., Foliente, G., Lupisek, A., Lützkendorf, T., Malmqvist, T., Garcia Martinez, A., Ouellet-Plamondon, C., Passer, A., Peuportier, B., Ramseier, L. and Satola, D., 2022. Existing benchmark systems for assessing global warming potential of buildings – Analysis of IEA EBC Annex 72 cases. *IOP Conference Series: Earth and Environmental Science*, 1078(012054), s. 1-11.
doi: 10.1088/1755-1315/1078/1/012054.

Santos, R., Costa, A., Silvestre, J. & Pyl, L., 2019. Informetric analysis and review of literature on the role of BIM in sustainable construction. *Automation in Construction*. 103(1), s. 221-234.
doi: 10.1016/j.autcon.2019.02.022.

Santos, R., Costa, A., Silvestre, J. & Pyl, L., 2019. Integration of LCA and LCC analysis within a BIM-based environment. *Automation in Construction*. 103(1), s. 127-149. doi: 10.1016/j.autcon.2019.02.011.

- Serrano-Baena, M.M., Ruiz-Díaz, C., Pablo Gilabert Boronat & Mercader-Moyano, P., 2023. Optimising LCA in complex buildings with MLCAQ: A BIM-based methodology for automated multi-criteria materials selection. *Energy and Buildings*. 294(113219), s. 1-12. doi: 10.1016/j.enbuild.2023.113219.
- Soust-Verdaguer, B., Llatas, C. & García-Martínez, A., 2017. Critical review of bim-based LCA method to buildings. *Energy and Buildings*. 136(1), s. 110-120. doi: 10.1016/j.enbuild.2016.12.009.
- Tam, V.WY., Zhou, Y., Shen, L. & Le, K.N., 2023. Optimal BIM and LCA integration approach for embodied environmental impact assessment. *Journal of Cleaner Production*. 385(135605), s. 1-15. doi: 10.1016/j.jclepro.2022.135605.
- Teng, Y., Xu, J., Pan, W. & Zhang, Y., 2022. A systematic review of the integration of building information modeling into life cycle assessment. *Building and Environment*. 221(109260), s. 1-13. doi: 10.1016/j.buildenv.2022.109260.
- Theißen, S., Hoper, J., Drzymalla, J., Wimmer, R., Markova, S., Meins-Becker, A. & Lambertz, M., 2020. Using Open BIM and IFC to Enable a Comprehensive Consideration of Building Services within a Whole-Building LCA. *Sustainability*. 12(5644), s. 1-25. doi: 10.3390/su12145644.
- Theißen, S., Hoper, J., Drzymalla, J., Wimmer, R., Zibell, M., Meins-Becker, A., Rössig, S., Goitowski, S. & Lambertz, M., 2020. BIM integrated automation of whole building life cycle assessment using German LCA data base ÖKOBAUDAT and Industry Foundation Classes. *IOP Conf. Series: Earth and Environmental Science*. 588(032025), s. 1-10. doi: 10.1088/1755-1315/588/3/032025.
- Tucker, S., Ambrose, M., Johnston, D., Newton, P., Seo, S. & Jones, D., 2003. LCADesign: An integrated approach to automatic eco-efficiency assessment of commercial buildings. *Construction Informatics Digital Library*. [Online]. s. 1-10. [Lokaliseret 12.11.2023].
Tilgængelig fra: <https://itc.scix.net/pdfs/w78-2003-403.content.pdf>.
- van Berlo, L., 2019. The curious case of the MVD. buildingSMART international. [Online].
[Lokaliseret 26.12.2023]. Tilgængelig fra: <https://www.buildingsmart.org/the-curious-case-of-the-mvd/>.
- Wastiels, L. & Decuyper, R., 2019. Identification and comparison of LCA-BIM integration strategies. *IOP Conf. Series: Earth and Environmental Science*. 323(012101), s. 1-8. doi: 10.1088/1755-1315/323/1/012101.
- Wohlin, C., 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. *EASE '14: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. 38(1), s. 1-10. doi: 10.1145/2601248.2601268.
- Woodward, D.G., 1997. Life cycle costing—Theory, information acquisition and application. *International Journal of Project Management*. 15(6), s. 335-344. doi: 10.1016/s0263-7863(96)00089-0.
- Xu, J., Teng, Y. & Pan, W., 2021. A BIM-LCA Integrated Method for Enhancing Efficiency of Embodied Carbon Estimation of Prefabricated High-Rise Buildings. *Proceedings of 37th Annual ARCOM Conference, 6-7 September 2021*. UK, Association of Researchers in Construction Management, s.14-23.

Xu, J., Teng, Y., Pan, W. & Zhang, Y., 2022. BIM-integrated LCA to automate embodied carbon assessment of prefabricated buildings. *Journal of Cleaner Production*. 374(133894), s. 1-16. doi: 10.1016/j.jclepro.2022.133894.

Xu, Z., Wang, S. & Wang, E., 2019. Integration of BIM and Energy Consumption Modelling for Manufacturing Prefabricated Components: A Case Study in China. *Advances in Civil Engineering*. 2019(1609523), s. 1-18. doi: 10.1155/2019/1609523.

Zhang, S.D., Zhang, S., Wu, Z., Wang, X., Jiang, Z., Wang, C. & Zhao G., 2023. IFC-enabled LCA for carbon assessment in pumped storage hydropower (PSH) with concrete face rockfill dams. *Automation in Construction*. 156(105121), s. 1-18. doi: 10.1016/j.autcon.2023.105121.

Zheng, B., Hussain, M., Yang, Y., Chan, A.P.C. & Chi, H.-L., 2023. Trade-offs between accuracy and efficiency in BIM-LCA integration. *Engineering, Construction and Architectural Management*. s. 1-22. doi: 10.1108/ecam-03-2023-0270.

Zimmermann, R.K., Bruhn, S. & Birgisdóttir, H., 2021. BIM-Based Life Cycle Assessment of Buildings - An Investigation of Industry Practice and Needs. *Sustainability*. 13(5455), s. 1-21. doi: 10.3390/su13105455.