

Bijlage 2:

Verdiepende toelichting deelvraag 3 en 5

Inhoud

1. Aanvulling op onderzoeksresultaten.....	2
1.1. Deelvraag 3: Welk van de standaarden voor elementenbegrotingen, STABU of NL-SfB, past het best bij Domera?	2
1.1.1. Domera	2
1.1.2. Extra toelichting op de multicriteria analyse.....	3
1.2. Deelvraag 5: Hoe kan een realtime koppeling tussen een 3D model en het kostenramingssysteem bij Domera wijzigingen automatisch en foutloos doorvoeren in begrotingen en offertes?	5
1.2.1. Gemelde problemen/bugs.....	5
1.2.2. Uitkomst poging 1.....	6
1.2.3. Uitkomst poging 2	8
1.2.3.1. Offerte	9
1.2.3.2. Begroting	15
Literatuurlijst	19

1. Aanvulling op onderzoeksresultaten

1.1. Deelvraag 3: Welk van de standaarden voor elementenbegrotingen, STABU of NL-SfB, past het best bij Domera?

1.1.1. Domera

Domera, opgericht door Maarten en Marius Vroegindeweij, is gespecialiseerd in de ontwikkeling van duurzame drijvende woningen. Maarten, bouwkundig ingenieur, brengt expertise in 3D BIM, robotisering en automatisering in, terwijl Marius, met ervaring als aannemer en projectleider, de verbinding legt tussen theorie en praktijk. Samen richten zij zich op CO2 neutrale woningen, waarbij technologieën zoals robotisering en Virtual Reality (VR) worden ingezet.

Het aanbod varieert van luxe drijvende woningen tot tiny homes, die optimaal gebruik maken van het beschikbare wateroppervlak in Nederland. Alle woningen zijn gasloos en 'nul op de meter', aangedreven door zonnepanelen. Door prefab technieken zoals houtskeletbouw (HSB) en cross laminated timber (CLT) wordt materiaalverspilling beperkt en de bouw efficiënt uitgevoerd. Domera onderscheidt zich door flexibiliteit in ontwerp, ongehinderd door de beperkingen van sluizenbreedtes.

Met behulp van 3D modellen genereert Domera open begrotingen, die transparantie bieden in kosten. Virtual en Augmented Reality (AR) stellen klanten in staat hun woning in detail te ervaren en ontwerpen naar wens aan te passen. Robotisering versnelt het bouwproces en verhoogt de efficiëntie.

Vanaf de productielocatie aan de Hollandsche IJssel in Gouda, op een historische scheepswerf (zie figuur 1), worden de woningen direct te water gelaten en vervoerd naar hun bestemming. Het jonge team van Domera combineert innovatie en een passie voor waterwonen om de droom van duurzaam wonen op het water werkelijkheid te maken (Over Domera – Domera, z.d.).



Figuur 1, historische scheepswerf

1.1.2. Extra toelichting op de multicriteria analyse

De multicriteria analyse (MCA) die in de scriptie is opgenomen, vergelijkt de twee classificatiesystemen STABU en NL-SfB aan de hand van zes zorgvuldig geselecteerde criteria. Deze criteria zijn gekozen om te evalueren welk systeem het beste aansluit bij de specifieke behoeften van Domera en hun BIM-processen. De scores voor elk systeem zijn gebaseerd op de mate van vervulling van de eisen binnen elk criterium, resulterend in een totaalscore die aangeeft welk systeem de meest geschikte keuze is voor toepassing binnen Domera. Hieronder volgt een gedetailleerde toelichting per criterium.

Criteria	Weging	STABU	NL-SfB
BIM Integratie	30%	3	4
Detailniveau van technische eisen	25%	4	3
Flexibiliteit in gebruik	20%	3	4
Gebruiksgemak	10%	2	3
Ondersteuning van calculaties	10%	3	4
Ondersteuning van planning	5%	3	4
Totaalscore	100%	3,20	3,75

Criteria	Toelichting
BIM Integratie	Hoe goed integreert het systeem met bestaande BIM-processen?
Detailniveau van technische eisen	Helpt het systeem om gedetailleerde specificaties vast te leggen?
Flexibiliteit in gebruik	Kan het systeem worden aangepast aan specifieke projectvereisten?
Gebruiksgemak	Hoe intuïtief en gemakkelijk is het om te implementeren?
Ondersteuning van calculaties	Ondersteunt het systeem het automatisch verwerken van gegevens in calculaties?
Ondersteuning van planning	Kan het systeem bijdragen aan efficiënte projectplanning?

BIM Integratie (30%)

Dit criterium beoordeelt hoe goed elk systeem integreert met bestaande BIM-processen, een essentiële factor voor Domera, aangezien het bedrijf werkt met BIM-technologie voor zowel projectbeheer als -visualisatie.

- **STABU (Score 3):** STABU biedt redelijke integratie met BIM-processen, maar de structuur van bouwdelen is minder flexibel, wat de integratie in verschillende fasen van het bouwproces bemoeilijkt. De focus van STABU ligt voornamelijk op gedetailleerde productcatalogi, wat de toepassing in de vroege ontwerpfases bemoeilijkt;
- **NL-SfB (Score 4):** NL-SfB scoort hoger op dit criterium vanwege de flexibele structuur, die uitstekend toepasbaar is in alle fasen van het bouwproces, van ontwerp tot uitvoering. De betere afstemming op BIM-modellen maakt het systeem dynamisch inzetbaar gedurende de verschillende ontwikkelingsfasen van projecten.

Detailniveau van technische eisen (25%)

Dit criterium beoordeelt het vermogen van het systeem om gedetailleerde technische eisen vast te leggen, wat essentieel is voor het opstellen van juridische en technische specificaties.

- **STABU (Score 4):** STABU biedt gedetailleerde en specifieke informatie, die zowel voor technische als juridische eisen geschikt is. Dit maakt het systeem bij uitstek geschikt voor complexe projecten waarbij gedetailleerde specificaties van groot belang zijn;
- **NL-SfB (Score 3):** NL-SfB biedt minder gedetailleerde informatie, wat het systeem minder geschikt maakt voor juridische specificaties. Het richt zich meer op de bouwdeelstructuur en hoewel dit nuttig is voor de coördinatie van werkzaamheden, mist het de gedetailleerdheid die nodig is voor formele documentatie.

Flexibiliteit in gebruik (20%)

Dit criterium meet in hoeverre het systeem kan worden aangepast aan specifieke projectbehoeften, wat van groot belang is voor Domera om hun werkprocessen efficiënt in te richten.

- **STABU (Score 3):** STABU biedt redelijke flexibiliteit in de bouwfase, maar mist de benodigde aanpasbaarheid in latere fasen van het project. De structuur van STABU is vooral gericht op de bouwtechnische specificaties, wat de toepasbaarheid in andere fasen beperkt;
- **NL-SfB (Score 4):** NL-SfB is bijzonder flexibel en kan door het gehele bouwproces effectief worden toegepast, van de ontwerpfase tot uitvoering en onderhoud. De gestructureerde aanpak maakt het gemakkelijk aan te passen aan de specifieke eisen van elk project.

Gebruiksgemak (10%)

Dit criterium beoordeelt hoe eenvoudig het systeem is te implementeren en te gebruiken, wat de snelheid van adoptie en efficiëntie in het dagelijks gebruik beïnvloedt.

- **STABU (Score 2):** STABU is complexer en vereist een langere leercurve, wat de initiële implementatie bemoeilijkt. Het gedetailleerde karakter van het systeem vormt een uitdaging voor gebruikers die niet vertrouwd zijn met de structuur;
- **NL-SfB (Score 3):** NL-SfB is gebruiksvriendelijker door de gestructureerde elementcoderingen. Dit vergemakkelijkt het begrip en de toepassing, verkort de implementatietijd en verhoogt de efficiëntie.

Ondersteuning van calculaties (10%)

Dit criterium beoordeelt in hoeverre het systeem geautomatiseerde gegevensverwerking ondersteunt voor kostencalculaties, wat cruciaal is voor efficiënt budgetbeheer.

- **STABU (Score 3):** STABU biedt redelijke ondersteuning voor calculaties, maar de automatisering is minder geavanceerd dan bij NL-SfB. Het systeem vereist meer handmatige input en biedt beperkte mogelijkheden voor geautomatiseerde prijsberekeningen;
- **NL-SfB (Score 4):** NL-SfB biedt robuuste ondersteuning voor automatisering in calculaties. De systematiek is goed geïntegreerd met kostencalculaties, wat snellere

en efficiëntere prijsberekeningen mogelijk maakt en bijdraagt aan een vlottere uitvoering van projecten.

Ondersteuning van planning (5%)


Dit criterium beoordeelt in hoeverre het systeem kan bijdragen aan het plannen van het bouwproces, bijvoorbeeld door fasering en tijdsplanning te ondersteunen.

- **STABU (Score 3):** STABU biedt beperkte ondersteuning voor planning. Het richt zich voornamelijk op het vastleggen van technische specificaties en bouwdelen, en biedt minder hulpmiddelen voor het beheer van de planning;
- **NL-SfB (Score 4):** NL-SfB biedt meer flexibiliteit op het gebied van planning. De structuur sluit goed aan bij de fasering van projecten, wat helpt bij het beheer van de tijdsplanning en de fasering van bouwactiviteiten.

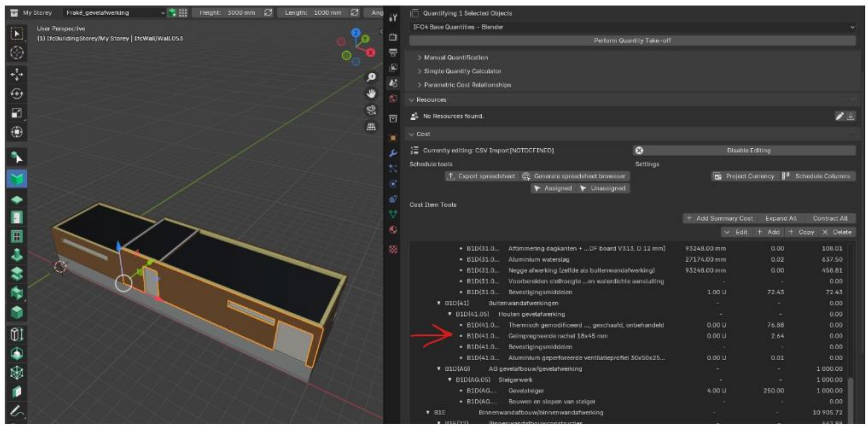
1.2. Deelvraag 5: Hoe kan een realtime koppeling tussen een 3D model en het kostenramingssysteem bij Domera wijzigingen automatisch en foutloos doorvoeren in begrotingen en offertes?

1.2.1. Gemelde problemen/bugs

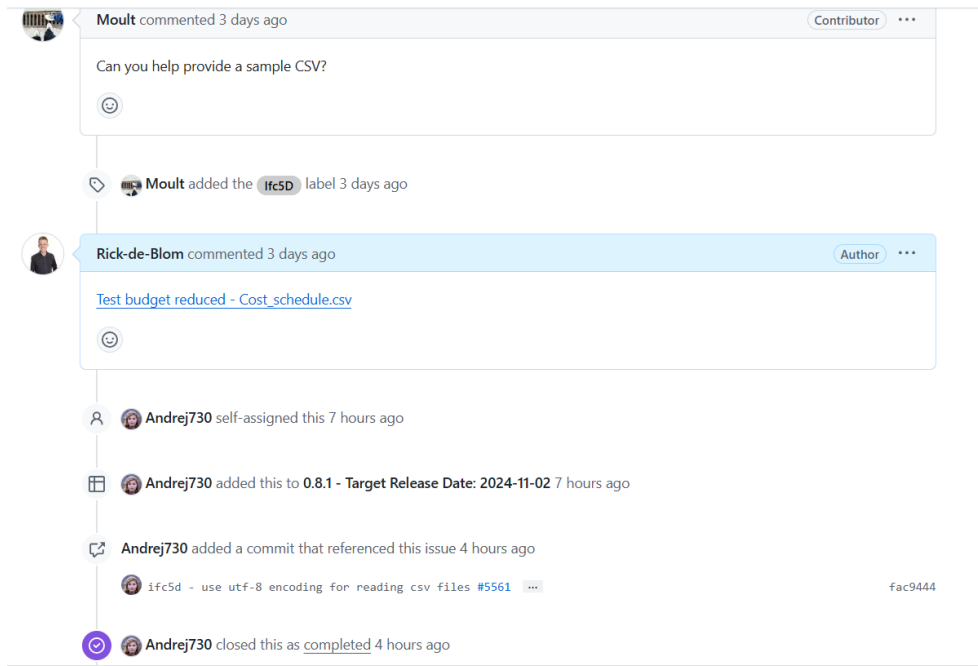
Op dit moment zijn er nog enkele knelpunten bij het weergeven van lengtes in meters binnen de begroting en het koppelen van een formule aan een kostenpost. Hiervoor is een [issue](#) aangemeld op OSArch en een [issue](#) op Github. Eén van de vragen is inmiddels opgelost, zoals hieronder weergegeven in figuur 2.

 Rick-de-Blom commented 3 days ago

When importing a CSV file that contains special characters such as accented letters (e.g., é, à, è) into the cost schedule of Bonsai, an error occurs. Specifically, rows containing these characters are left empty after the import, despite the fact that these characters work correctly when used directly in other parts of the system, such as the "Grouping and Filtering" search function and the "Spreadsheet Import/Export" feature. This issue seems isolated to CSV imports for the cost schedule.



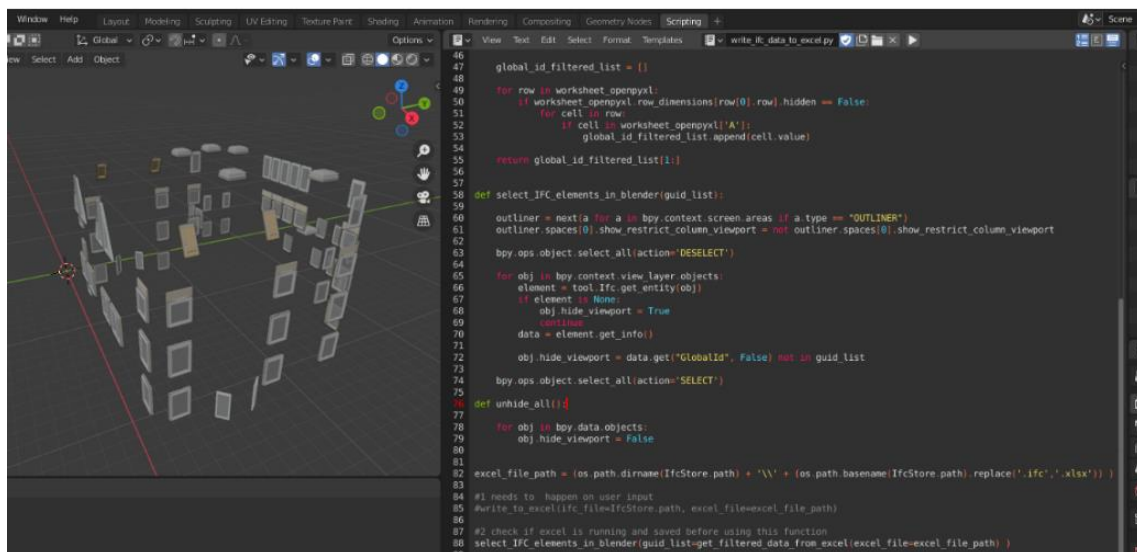
Cost Item Tools	Item	Unit	Quantity	Unit Price	Amount	Contract Price
• R10413.1	Aluminium ingangsbank - 100 x 120 x 120 mm	1.00 U	1.00	120.00	120.00	120.00
• R10413.2	Aluminium wandkast	2.71 x 1.00 mm	0.02	437.50	8.75	8.75
• R10413.3	Wegge afwerking (zelfde als buitenwandafwerking)	1.00 U	1.00	437.50	437.50	437.50
• R10413.4	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.5	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.6	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.7	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.8	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.9	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.10	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.11	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.12	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.13	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.14	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.15	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.16	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.17	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.18	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.19	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.20	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.21	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.22	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.23	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.24	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.25	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.26	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.27	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.28	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.29	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.30	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.31	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.32	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.33	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.34	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.35	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.36	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.37	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.38	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.39	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.40	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.41	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.42	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.43	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.44	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.45	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.46	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.47	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.48	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.49	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.50	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.51	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.52	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.53	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.54	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.55	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.56	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.57	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.58	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.59	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.60	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.61	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.62	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.63	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.64	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.65	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.66	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.67	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.68	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.69	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.70	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.71	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.72	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.73	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.74	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.75	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.76	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.77	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.78	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.79	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.80	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.81	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.82	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.83	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.84	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.85	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.86	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.87	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.88	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.89	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.90	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.91	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.92	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.93	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.94	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.95	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.96	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.97	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.98	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10413.99	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.00	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.01	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.02	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.03	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.04	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.05	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.06	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.07	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.08	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.09	Aluminium wandkast met 2x120 x 120 mm	1.00 U	1.00	437.50	437.50	437.50
• R10414.10	Aluminium wandkast met 2x120 x 120 mm	1				



Figuur 2, <https://github.com/lfcOpenShell/lfcOpenShell/issues/5561>

1.2.2. Uitkomst poging 1

In de eerste poging is geprobeerd om met behulp van een Python/lfcOpenShell code de begroting automatisch in te laden. Uit deze poging is gebleken dat stap 2, het klikken op “search”, niet geautomatiseerd kan worden binnen dit onderzoek. Verschillende personen hebben voor andere doeleinden een vergelijkbare poging ondernomen, waarbij uiteindelijk een code is ontwikkeld, zoals te zien is in figuur 3. Deze code vereist onder andere het inladen van een Excel lijst om bepaalde onderdelen op te halen. De complexiteit van de code is aanzienlijk, wat het minder gebruiksvriendelijk maakt, vooral bij toekomstige wijzigingen of prijsupdates. In figuur 4 hieronder is een deel van de code te zien die in deze poging is geschreven.



Figuur 3, (Blender BIM Hide Isolating in Context With Python, 2021)


```
1 import bpy
2 import ifcopenshell
3 import ifcopenshell.util.selector
4 import ifcopenshell.util.element
5
6 #updaten van eenheden
7 bpy.ops.bim.hotkey(hotkey="S_Q", description="bpy.ops.bim.perform_quantity_take_off()\nPerform a quantity
8
9 bpy.ops.object.select_all(action='DESELECT')
10
11
12 #naamgeving Cost Schedules en toevoegen arbeid
13 name = ("13.2")
14 number=0
15 bpy.context.scene.BIMCostProperties.cost_items[number].identification = name
16 bpy.context.scene.BIMCostProperties.cost_items[number].name = "Vloeren op grondslag; constructief"
17 number=number + 1
18 bpy.context.scene.BIMCostProperties.cost_items[number].identification = name
19 bpy.context.scene.BIMCostProperties.cost_items[number].name = "Betoncasco"
20
21 name = ("21.2")
22 number=number + 1
23 bpy.context.scene.BIMCostProperties.cost_items[number].identification = name
24 bpy.context.scene.BIMCostProperties.cost_items[number].name = "Buitenwanden; constructief"
25 number=number + 1
26 bpy.context.scene.BIMCostProperties.cost_items[number].identification = name
27 bpy.context.scene.BIMCostProperties.cost_items[number].name = "HSB wand 38x184"
28
29 number=number + 1
30 cost_value=2218
31 bpy.context.scene.BIMCostProperties.cost_items[number].identification = name
32 bpy.context.scene.BIMCostProperties.cost_items[number].name = "Geïmpregneerde rachels 18x45 mm"
33 bpy.ops.bim.enable_editing_cost_item_values(cost_item=81)
34 bpy.ops.bim.enable_editing_cost_item_value(cost_value=cost_value)
35 bpy.context.scene.BIMCostProperties.cost_value_attributes[2].float_value = (0.88*2.5)
36 bpy.ops.bim.edit_cost_value(cost_value=cost_value)
37 bpy.ops.bim.disable_editing_cost_item()
38 bpy.ops.bim.select_type(relating_type=448)
39 bpy.ops.bim.select_similar_type()
40 bpy.ops.bim.hotkey(hotkey="S_Q", description="bpy.ops.bim.perform_quantity_take off()\nPerform a quantity
41 bpy.ops.bim.assign_cost_item_quantity(cost_item=81, related_object_type="PRODUCT", prop_name="NetSideArea"
42 bpy.ops.object.select_all(action='DESELECT')
43
44 cost_value=2219
45 number=number + 1
```

```
1 import bpy
2 import ifcopenshell
3 import ifcopenshell.util.selector
4 import ifcopenshell.util.element
5
6 #openen van Cost Schedules en Work Schedules
7 bpy.ops.bim.enable_editing_cost_items(cost_schedule=66)
8
9 bpy.ops.bim.enable_editing_work_schedule_tasks(work_schedule=867)
10
11 bpy.ops.bim.load_resources()
12
13 bpy.context.scene.BIMResourceProperties.active_resource_index = 3
14
15 bpy.context.scene.BIMWorkScheduleProperties.should_show_column_ui = True
16 bpy.ops.bim.setup_default_task_columns()
17 bpy.context.scene.BIMWorkScheduleProperties.column_types = 'Special'
18 bpy.context.scene.BIMWorkScheduleProperties.other_columns = 'Controls.Calendar'
19 bpy.ops.bim.add_task_column(column_type="Controls", name="Calendar", data_type="string")
20 bpy.context.scene.BIMWorkScheduleProperties.should_show_column_ui = False
```

Figuur 4, (Deel geschreven code)

1.2.3. Uitkomst poging 2

In de tweede poging is onderzocht in hoeverre het mogelijk is een spreadsheet te creëren, deze te exporteren naar een CSV-bestand en deze vervolgens in te laden in Bonsai. Uit dit experiment blijkt dat deze methode haalbaar is. Binnen de spreadsheet is een standaard begroting opgezet waarin de naamgeving van elk type element per regel is vastgelegd. Bij het importeren van het CSV-bestand kan deze waarde automatisch worden uitgelezen en direct worden gekoppeld aan de juiste regel in Bonsai.

Hieronder volgt een gedetailleerde toelichting van de functies per kolom in de spreadsheet:

Hierarchy

Deze kolom biedt een hiërarchische structuur waarmee hoofdonderdelen, subonderdelen en details binnen de kostenstructuur worden geordend. Zo vertegenwoordigt een waarde van '1' een hoofdcategorie (bijvoorbeeld totale bouwkosten), terwijl subcategorieën zoals funderingen waarden als '2' of '3' hebben.

Identification

Deze kolom bevat classificatiestandaarden volgens de NL-SfB, gebaseerd op de NEN 2699:2017. Hiermee worden unieke codes gekoppeld aan specifieke bouwonderdelen.

Name

Deze kolom bevat een beschrijving van elk item, zoals "Totale bouwkosten" of "Fundering". Dit veld is essentieel voor de interpretatie door gebruikers.

Quantity

De hoeveelheden van elk onderdeel worden in deze kolom vastgelegd.

Unit

Hierin wordt de eenheid van de hoeveelheden aangegeven, bijvoorbeeld m² of m³.

Rate

Deze kolom blijft voorlopig leeg en wordt later ingevuld.

Contract Rate

Deze kolom bevat tarieven gerelateerd aan onderaanneming.

Material Rate

Tarieven die verband houden met materiaalgebruik worden hier vastgelegd.

Labor Rate

Deze kolom registreert tarieven die betrekking hebben op arbeidskosten.

Subtotal

Deze kolom wordt gebruikt voor tussenberekeningen en wordt automatisch ingevuld bij het importeren in Bonsai.

Total Price

De totaalprijs per onderdeel wordt in deze kolom weergegeven en eveneens automatisch berekend bij het importeren.

Property

In deze kolom wordt vastgelegd welke waarde moet worden uitgelezen, bijvoorbeeld netto of bruto en in welke eenheid (bijvoorbeeld lengte, m², m³).

Query

Deze kolom is gekoppeld aan de property kolom en bevat de gegevens of namen van de elementen die moeten worden uitgelezen.

Vervolgstappen

Op basis van deze bevindingen wordt het proces verder aangescherpt en geoptimaliseerd. Hieronder wordt het volledige proces voor het automatisch genereren van een offerte en bijbehorende begroting stapsgewijs uiteengezet.

1.2.3.1. Offerte

Het automatiseren van offertes binnen het bouwproces is een cruciaal aspect van efficiënt projectbeheer. Domera onderzoekt de mogelijkheden van Bonsai om dergelijke processen te stroomlijnen, met specifieke aandacht voor het dynamisch genereren van documenten op basis van BIM-data.

In dit kader is binnen dit onderzoek een script ontwikkeld dat, met behulp van de Python-console van Bonsai, gegevens uit een IFC-bestand extraheren en verwerken. Dit script combineert functionaliteiten van diverse bronnen zoals IfcOpenShell en PDFKit, en integreert beeld- en tekstinformatie in een gestructureerd formaat. Het uiteindelijke resultaat is een geautomatiseerde PDF-uitvoer, geschikt voor offertes.

De volgende secties beschrijven de opzet en werking van het script, van het importeren van modules en het instellen van paden tot het genereren van HTML-content en de conversie naar een PDF-document. Door deze werkwijze te documenteren, wordt een solide basis gelegd voor verdere optimalisatie van het proces, waarbij de nadruk ligt op snelheid, betrouwbaarheid en gebruiksgemak.

Eerste opzet code

In de Python console van Bonsai is een deel van de code geschreven op basis van de functionaliteiten van (ChatGPT, z.d.) met behulp van (W3Schools.com, z.d.) en (IfcOpenShell 0.8.0 Documentation, z.d.). Hieronder in figuur 5 worden de commando's weergegeven die

zijn gebruikt om dit proces te genereren. Deze basis vormt de grondslag voor de verdere uitwerking van de code.

Is het mogelijk om in Bonsai (voorheen BlenderBIM) een afbeelding in te laden voor een offerte?

Ja, in **Bonsai** (voorheen **BlenderBIM**) is het mogelijk om afbeeldingen toe te voegen aan een offerte, hoewel dit niet standaard onderdeel is van de kernfunctionaliteit van het programma. In veel BIM-gerelateerde toepassingen, waaronder BlenderBIM/Bonsai, kunnen afbeeldingen of grafische elementen worden ingevoegd als visuele ondersteuning binnen rapporten, tekeningen of presentaties.

Er zijn verschillende manieren waarop afbeeldingen geïntegreerd kunnen worden in een offerte, bijvoorbeeld door:

1. **Toevoegen van afbeeldingen in de Blender-omgeving:** Blender zelf biedt de mogelijkheid om afbeeldingen te importeren en te gebruiken als referentie of onderdeel van de 3D-scène. Deze afbeeldingen kunnen dienen als visuele ondersteuning voor de tekening of het model dat in de offerte wordt gebruikt.

Geef mij voor stap 2.1 een volledige uitwerking wat zou kunnen werken in Bonsai

Een volledige uitwerking voor stap 2.1 in Bonsai zou bestaan uit het ontwikkelen van een Python-script dat de IFC-data uitleest, de relevante gegevens herkent (zoals het type wand en de afwerking), en vervolgens een afbeelding toevoegt aan het rapport (bijvoorbeeld een PDF). Hier is een voorbeeld van hoe dit eruit zou kunnen zien.

Volledig Python-script voor het integreren van afbeeldingen in Bonsai:

1. Voorbereidingen

- Zorg dat je Bonsai en BlenderBIM geïnstalleerd hebt.
- Installeer extra benodigde bibliotheken voor het genereren van PDF's, zoals `reportlab`.
- Zorg ervoor dat de afbeelding van de Frake gevelafwerking beschikbaar is in je bestandssysteem (bijvoorbeeld als `Frake_gevelafwerking.jpg`).



Ik kan geen reportlab toevoegen aan Bonsai, zijn er andere mogelijkheden?

Als je geen extra Python-bibliotheken zoals `reportlab` kunt toevoegen aan Bonsai, zijn er andere manieren om afbeeldingen te integreren in een rapport of offerte. Je kunt bijvoorbeeld gebruikmaken van bestaande mogelijkheden in Bonsai en BlenderBIM of van standaard Python-tools die geen extra installaties vereisen.

Hier zijn enkele alternatieve benaderingen:

1. Blender's eigen renderfunctionaliteit:

Blender biedt ingebouwde render- en tekstmogelijkheden. Je kunt Blender's **Python API** gebruiken om de afbeelding van de Frake-gevelafwerking in je 3D-scène te plaatsen en vervolgens een render of een screenshot te maken die je aan de offerte toevoegt. Deze methode vermijdt de noodzaak van externe bibliotheken en integreert naadloos in de Bonsai/BlenderBIM-omgeving.

De HTML code werkt goed! Is hier ook nog een mogelijkheid dat het automatisch uitgelezen wordt aan de hand van gekoppelde elementen in de begrotingsfunctie van Bonsai?

Ja, het is mogelijk om de HTML-code automatisch te genereren op basis van gekoppelde elementen in de begrotingsfunctie van Bonsai. Dit kan door gebruik te maken van een script dat de IFC-data uitleest, de gekoppelde elementen herkent (zoals types en afwerkingen) en vervolgens de HTML-inhoud dynamisch genereert op basis van deze gegevens.

Stappen om de HTML automatisch te genereren op basis van IFC-gegevens:

1. **Lees IFC-bestand uit:** Het script leest de IFC-data om te controleren welke elementen, zoals `IfcWallType` en de bijbehorende afwerkingen, aanwezig zijn.
2. **Controleer eigenschappen van elementen:** Het script zoekt naar specifieke eigenschappen van elk element (bijvoorbeeld de afwerking "Frake") en koppelt deze aan relevante informatie (zoals kosten, afbeeldingen).

Figuur 5, commando's binnen ChatGPT

Overzicht van het script

Dit script is ontworpen om gegevens te extraheren uit een IFC-bestand en deze te gebruiken voor het genereren van een offerte. De offerte wordt uiteindelijk geëxporteerd als een PDF-document. De kernfunctionaliteit omvat:

- Het laden en lezen van een IFC-bestand;
- Het coderen van afbeeldingen naar Base64 voor gebruik in HTML-documenten;
- Het dynamisch genereren van een HTML-document op basis van elementen in het IFC-model;
- Het exporteren van de resultaten naar een PDF-bestand in een gespecificeerde map.

1. Importeren van modules

Het script begint met het importeren van verschillende Python modules:

- **Ifcopenshell**: Voor het openen en uitlezen van gegevens uit IFC-bestanden;
- **PdfKit**: Voor de conversie van HTML-documenten naar PDF;
- **Base64**: Voor het omzetten van afbeeldingen naar base64 code, wat vereist is voor het embedden in HTML;
- **Os** en **sys**: Voor bestands- en padbeheer.

Deze modules vormen de kern van het script, omdat ze de functionaliteiten bieden om bouw informatie te verwerken en te presenteren.

```
1 import ifcopenshell
2 import pdfkit
3 import base64
4 import os
5 import sys
```

2. Instellen van paden

De paden naar belangrijke bestanden en directories worden dynamisch ingesteld:

- **Basispad**: Het basepath geeft aan waar bestanden zoals het IFC-bestand en de benodigde code zich bevinden;
- **IFC bestandspad**: Dit pad verwijst naar het specifieke IFC-bestand dat de bron van data vormt. Het bestand wordt geladen met ifcopenshell.open;
- **Codepad**: Voor het organiseren van scriptmodules of aanvullende functionaliteit.

Hierdoor kan het script flexibel werken met verschillende projectstructuren.

```
6
7 # Basispad
8 basepath = "C:/Users/rickd/OneDrive/Desktop/Projecten/Code"
9
10 # IFC-bestandspad
11 ifc_file_path = os.path.join(basepath, "Begroting_Gouda.ifc")
12 # Laad het IFC-bestand
13 ifc_file = ifcopenshell.open(ifc_file_path)
14
15 # Voeg paden voor geïmporteerde modules toe
16 code_path = os.path.join(basepath, "Code")
17
```

3. Afbeeldingen omzetten naar base64

De functie `encode_image_to_base64` converteert afbeeldingsbestanden naar base64 gecodeerde strings:

- **Voordelen van base64**: Afbeeldingen kunnen rechtstreeks in HTML worden opgenomen, zonder afhankelijk te zijn van externe bestanden;
- **Caching**: Een interne `image_cache` zorgt ervoor dat dezelfde afbeelding niet meerdere keren wordt verwerkt, wat de efficiëntie verhoogt;

- **Foutafhandeling:** Als een afbeeldingsbestand niet wordt gevonden, retourneert de functie None. Dit voorkomt dat het script crasht bij ontbrekende afbeeldingen.

```

20 def encode_image_to_base64(image_path):
21     """Encodeer afbeelding naar base64 voor HTML."""
22     if image_path in image_cache:
23         return image_cache[image_path]
24
25     try:
26         with open(image_path, "rb") as image_file:
27             encoded = base64.b64encode(image_file.read()).decode('utf-8')
28             image_cache[image_path] = encoded # Sla het gecodeerde resultaat op
29         return encoded
30     except FileNotFoundError:
31         return None # Als afbeelding niet bestaat, retourneer None

```

4. HTML-content genereren

De functie `generate_html_content` genereert een HTML-template voor elk bouwkundig element:

- **Elementnaam:** De naam van het bouwonderdeel wordt als titel gebruikt;
- **Beschrijving van materiaalagen:** Zowel de algemene beschrijving van een laagset als de beschrijvingen van individuele lagen worden weergegeven;
- **Afbeelding:** De afbeelding van het element wordt toegevoegd in base64 formaat.

Hierdoor ontstaat een gestructureerd overzicht dat zowel tekstuele als visuele informatie bevat.

```

33 def generate_html_content(element_name, layer_set_description, layer_descriptions, image_path):
34     """Genereer de HTML-content voor een element met beschrijving en afbeelding."""
35     base64_image = encode_image_to_base64(image_path) if image_path else None
36
37     # Voor de beschrijving van de MaterialLayerSet zonder bullet en met nieuwe regels na punten
38     if layer_set_description:
39         layer_set_desc_html = "<div>" + "</div><div>".join([desc.strip() for desc in layer_set_description.split(".") if desc.strip()]) + "
40     else:
41         layer_set_desc_html = ""
42
43     # Voor de beschrijvingen van de individuele lagen
44     description_list = ""
45     for desc in layer_descriptions:
46         for sub_desc in desc.split("."):
47             if sub_desc.strip():
48                 description_list += "<li>" + sub_desc.strip() + "</li>"
49
50     return f"""
51     <tr>
52         <td style='padding-left: 10px; padding-right: 10px; vertical-align: top;'>{element_name}</td>
53         <td style='vertical-align: top;'>
54             {layer_set_desc_html}
55             <ul style='padding-left: 15px; margin-right: 10px;'>{description_list}</ul>
56         </td>
57         <td style='padding: 10px; text-align: center;'>
58             {f"<img src='data:image/jpeg;base64,{base64_image}' alt='{element_name}' width='150'/">" if base64_image else "Afbeelding ni
59         </td>
60     </tr>

```

5. Genereren en opslaan van PDF

Met behulp van PDFKit wordt de gegenereerde HTML-content omgezet naar een PDF-document:

- **Standaardopties:** De conversie gebruikt standaard PDF-instellingen, maar deze kunnen worden aangepast voor specifieke lay-outvereisten;
- **Outputlocatie:** De PDF wordt opgeslagen in een vooraf gedefinieerde map. Dit maakt het eenvoudig om de gegenereerde offerte terug te vinden.

```

482
483 # Configuratie voor PDFKit
484 config = pdfkit.configuration(wkhtmltopdf="C:/Program Files/wkhtmltopdf/bin/wkhtmltopdf.exe")
485
486 # Probeer de PDF te genereren
487 pdf_path = os.path.join(basepath, "offerte.pdf")
488 try:
489     pdfkit.from_file(html_path, pdf_path, configuration=config)
490     print("PDF is gegenereerd:", pdf_path)
491 except Exception as e:
492     print("Fout bij het genereren van de PDF:", str(e))
  
```

6. Toepassing op IFC-data

Het script doorloopt bouwkundige elementen in het IFC-bestand:

- **Uitlezen van data:** Voor elk element worden relevante gegevens zoals namen, beschrijvingen en materiaallagen opgehaald;
- **Afbeeldingen koppelen:** Indien beschikbaar, worden afbeeldingen gekoppeld aan de elementen;
- **HTML-output genereren:** De data worden verwerkt tot een HTML-blok per element.

Dit proces zorgt ervoor dat de volledige bouwkundige dataset wordt vertaald naar een overzichtelijke offerte.

```

61 def get_property_set_values(ifc_file, property_set_name):
62     """Haal de eigenschapswaarden uit een PropertySet."""
63     property_values = []
64     for entity in ifc_file.by_type('IfcPropertySet'):
65         if entity.Name == property_set_name:
66             for prop in entity.HasProperties:
67                 property_name = prop.Name
68                 property_value = prop.NominalValue.wrappedValue if hasattr(prop.NominalValue, 'wrappedValue') else None
69                 if property_value:
70                     property_values.append((property_name, property_value))
71     return property_values
72
73 def generate_section_table(property_set_name, ifc_file, finishes):
74     """Genereer HTML-tabel voor een bepaalde PropertySet."""
75     property_values = get_property_set_values(ifc_file, property_set_name)
76     table_content = ""
77
78     for property_name, property_value in property_values:
79         beschrijvingen = [beschrijving.strip() for beschrijving in str(property_value).split(".") if beschrijving.strip()]
80         beschrijving_html = "<ul>" + "".join(f"<li>{beschrijving}</li>" for beschrijving in beschrijvingen) + "</ul>"
81
82         image_path = finishes.get(property_name)
83         base64_image = encode_image_to_base64(image_path) if image_path else None
84         afbeelding_html = f"<img src='data:image/jpeg;base64,{base64_image}' width='150' />" if base64_image else ""
85
86         table_content += f"""
87         <tr>
88             <td>{property_name}</td>
89             <td>{beschrijving_html}</td>
90             <td>{afbeelding_html}</td>
91         </tr>
92         """
93     return table_content
94
95 # Definieer de afwerkingen en hun bijbehorende afbeeldingen
96 sys.path.append(code_path)
97 from finishes import finishes
98 for element, path in finishes.items():
99     print(f"Element: {element}, Path: {path}")
100
  
```

Voor de volledige code, zie bijlage 6 “Code_HTML script voor offerte met PDF export in PDF”.

1.2.3.2. Begroting

Het opstellen van een nauwkeurige en gestructureerde begroting is een essentieel onderdeel van elk bouwproject. Binnen Domera wordt onderzocht hoe dit proces met behulp van Bonsai verder geautomatiseerd kan worden, zodat gegevens uit een IFC-model direct kunnen worden omgezet in een kostenraming. Hierbij ligt de focus op het efficiënt integreren van arbeids- en materiaalkosten in een dynamisch gegenereerd document, dat als PDF geëxporteerd kan worden.

In dit onderzoek is een script ontwikkeld dat via de Python-console van Bonsai diverse processen uitvoert, zoals het extraheren van gegevens uit `IfcResourceTime` en `IfcCostItem` entiteiten en het berekenen van arbeids- en materiaalkosten. Deze informatie wordt vervolgens gecombineerd in een HTML-document, dat dient als basis voor een PDF-export. Met behulp van modules zoals `IfcOpenShell` en `PDFKit` wordt een workflow gerealiseerd die zowel nauwkeurig als flexibel is.

De volgende secties beschrijven de opzet en werking van het script, waaronder het importeren van modules, het instellen van paden en het berekenen van kosten op basis van BIM-data. Door deze werkwijze te documenteren, wordt een robuuste basis gelegd voor het automatiseren van begrotingen, waarmee tijdswinst en betrouwbaarheid worden bevorderd.

Eerste opzet code

In de Python console van Bonsai is een deel van de code geschreven op basis van de functionaliteiten van (ChatGPT, z.d.) met behulp van (W3Schools.com, z.d.) en (`IfcOpenShell` 0.8.0 Documentation, z.d.). Hieronder in figuur 6 wordt de commando weergegeven die is gebruikt om dit proces te genereren. Deze basis vormt de grondslag voor de verdere uitwerking van de code.

Binnen Bonsai/Blender wil ik `CostItem` regels uitlezen en toevoegen aan mijn code. Hoe kan ik dit doen



Om `CostItem`-regels in Bonsai/BlenderBIM te kunnen uitlezen en toe te voegen, kun je gebruik maken van de uitgebreide functionaliteit die IFC ondersteunt voor het beheren van kosteninformatie in je model. Hier is een stapsgewijze aanpak om deze regels te benaderen en te integreren in je code:

1. **Verwerk de IFC-structuur:** `CostItems` kunnen worden gestructureerd als `IfcCostItem` entiteiten binnen je IFC-model. Dit zijn objecten die gekoppeld kunnen worden aan elementen, activiteiten, of andere `CostItems` via relaties zoals `IfcRelAssignsToControl`. Zorg ervoor dat je de juiste context en connecties binnen je IFC-model kent.
2. **Gebruik Python en BlenderBIM API:**
 - Zorg ervoor dat je een script in Blender schrijft dat de IFC-bestandstructuur kan doorlopen.
 - Je kunt de `IfcOpenShell` bibliotheek gebruiken om je IFC-model in te laden en objecten zoals `IfcCostItem` te benaderen. ↓

Figuur 6, commando binnen ChatGPT

Overzicht van het script

Dit script is ontworpen om gegevens te extraheren uit een IFC-bestand en deze te gebruiken voor het genereren van een kostenraming. De raming wordt uiteindelijk geëxporteerd als een PDF-document. De kernfunctionaliteit omvat:

- Het laden en lezen van een IFC-bestand;
- Het berekenen van arbeidskosten op basis van informatie uit IfcResourceTime entiteiten;
- Het ophalen en verwerken van kosteninformatie, zoals arbeids- en materiaalkosten, via IfcCostItem entiteiten;
- Het dynamisch genereren van een HTML-document als basis voor de PDF;
- Het exporteren van de resultaten naar een gespecificeerde map.

1. Importeren van modules

Het script begint met het importeren van essentiële modules:

- **Os:** Voor het manipuleren van padnamen;
- **Ifcopenshell:** Voor het openen en uitlezen van gegevens uit IFC-bestanden;
- **Pdfkit:** Voor de conversie van HTML-documenten naar PDF;
- **Locale:** Voor de juiste nummernotatie (bijvoorbeeld Nederlandse notatie met komma's in plaats van punten);
- **Sys:** Voor padbeheer.

```
1 import os
2 import ifcopenshell
3 import pdfkit
4 import locale
5 import sys
```

2. Instellen van de locale

De locale wordt ingesteld op nl_NL.UTF-8, wat nodig is voor een correcte weergave van getallen in de Nederlandse context.

```
7 # Stel de locale in voor de Nederlandse weergave van getallen
8 locale.setlocale(locale.LC_NUMERIC, 'nl_NL.UTF-8')
```

3. Definiëren van paden

Het script definieert een basispad (basepath) en de locatie van het IFC-bestand. Vervolgens wordt het IFC-bestand geopend met behulp van ifcopenshell.open.

```

10 # Basispad
11 basepath = "C:/Users/rickd/..."
12
13 # IFC-bestandspad
14 ifc_file_path = os.path.join(basepath, "Begroting_Gouda.ifc")
15 # Laad het IFC-bestand
16 ifc_file = ifcopenshell.open(ifc_file_path)
17
18 # Voeg paden voor geïmporteerde modules toe
19 code_path = os.path.join(basepath, "Code")
20

```

4. Functie: Extractie van arbeidstijden

Een functie genaamd `extract_resource_time` verwerkt alle entiteiten van het type `IfcResourceTime`. Voor elke entiteit worden de volgende stappen uitgevoerd:

- De `ScheduleWork` eigenschap wordt uitgelezen;
- Het aantal uren wordt berekend door de tijdsnotatie (PT...H) te parsen;
- De totale arbeidskosten worden berekend door het aantal uren te vermenigvuldigen met een uurtarief (hier 58 euro per uur).

```

21 def extract_resource_time(ifc_file):
22     resource_times = ifc_file.by_type("IfcResourceTime")
23     labor_costs = []
24
25     for resource_time in resource_times:
26         schedule_work = getattr(resource_time, "ScheduleWork", None)
27         hourly_rate = 58.0 # EUR per uur
28
29         if schedule_work:
30             hours = float(schedule_work[2:-1]) # Verwijder 'PT' en de eenheid 'H'
31             total_cost = hours * hourly_rate
32             labor_costs.append((resource_time.id(), hours, hourly_rate, total_cost))
33
34     return labor_costs
35
36 # Roep de functie aan om arbeidskosten te berekenen
37 labor_costs = extract_resource_time(ifc_file)
38

```

5. Functie: Extractie van monetary measures

Een functie genaamd `extract_monetary_measures` verwerkt alle entiteiten van het type `IfcCostItem` in een IFC-bestand. Voor elke entiteit worden de volgende stappen uitgevoerd:

- **Naam ophalen:** De naam van de `IfcCostItem` wordt gelezen via de eigenschap `Name`. Indien deze ontbreekt, wordt de naam ingesteld op "Onbekend";
- **Controleren op kostenwaarden:** Als de `IfcCostItem` gekoppelde kostenwaarden heeft (`CostValues`), worden deze doorlopen;
- **Filteren op monetaire waarden:** Voor elke kostenwaarde die een `IfcCostValue` is, wordt gecontroleerd of de waarde (`AppliedValue`) van het type `IfcMonetaryMeasure` is;
- **Waarde extraheren:** Indien aan bovenstaande voorwaarden is voldaan, wordt de monetaire waarde (`wrappedValue`) opgehaald;

- **Opslag in lijst:** De naam van de `IfcCostItem` en de monetaire waarde worden als tuple toegevoegd aan een lijst.

```

39 def extract_monetary_measures(ifc_file):
40     cost_items = ifc_file.by_type("IfcCostItem")
41     monetary_values = []
42
43     for cost_item in cost_items:
44         cost_name = cost_item.Name if hasattr(cost_item, 'Name') else "Onbekend"
45
46         if hasattr(cost_item, "CostValues") and cost_item.CostValues:
47             for cost_value in cost_item.CostValues:
48                 if cost_value.is_a("IfcCostValue"):
49                     if cost_value.AppliedValue and cost_value.AppliedValue.is_a("IfcMonetaryMeasure"):
50                         monetary_measure = cost_value.AppliedValue.wrappedValue
51                         monetary_values.append((cost_name, monetary_measure))
52
53     return monetary_values
54
55 # Roep de functie aan om de monetaire waarden en koppelingen te extraheren
56 monetary_values = extract_monetary_measures(ifc_file)
57

```

6. Genereren van een HTML-document

Het script bevat functionaliteiten voor het dynamisch genereren van een HTML-document. Dit document wordt vervolgens gebruikt voor de PDF-export. De exacte details van deze sectie zijn verderop in de code te vinden.

```

272 # Genereer het begrotingsrapport
273 html_path = os.path.join(basepath, "begroting.html")
274 budget_report = generate_budget_report(ifc_file, labor_costs)
275
276 # Opslaan als HTML-bestand
277 with open(html_path, "w", encoding="utf-8") as html_file:
278     html_file.write(budget_report)
279

```

7. PDF-export

Met behulp van PDFKit wordt de HTML geconverteerd naar een PDF-bestand, dat vervolgens in een specifieke map wordt opgeslagen.

```

280 # HTML omzetten naar PDF
281 wkhtmltopdf_path = "C:/Program Files/wkhtmltopdf/bin/wkhtmltopdf.exe"
282 config = pdfkit.configuration(wkhtmltopdf=wkhtmltopdf_path)
283
284 pdf_path = os.path.join(basepath, "begroting.pdf")
285 pdfkit.from_file(html_path, pdf_path, configuration=config)

```

Voor de volledige code, zie bijlage "Code_HTML script voor begroting met PDF export in PDF".

Literatuurlijst

Blender BIM hide isolating in context with Python. (2021).

<https://community.osarch.org/discussion/767/blender-bim-hide-isolating-in-context-with-python>

CcO-Photographers. (z.d.). Gratis Afbeeldingen : aantal, geld, bedrijf, dichtbij, merk, doopvont, bank, illustratie, ontwerp, tekst, euro, schuld, munt, besparen, rekenmachine, munten, financiën, dataïaufnahme, begroting, effectenbeurs, bankbiljet, dollarbiljet, macro foto 3000x2000. Pxhere. <https://pxhere.com/nl/photo/1021973>

ChatGPT. (z.d.). <https://chatgpt.com/>

IfcOpenShell 0.8.0 documentation. (z.d.). <https://docs.ifcopenshell.org/>

Over Domera – domera. (z.d.). <https://domera.nl/over-domera/>

W3Schools.com. (z.d.). <https://www.w3schools.com/html/default.asp>