

Masterproef:

Virtueel ontwerpen met IFC Automatiseren meetstaat en kostprijsberekening

Kandidaat: Tim Lemoine

Promotoren

Interne promotor: ing. C. Vuye

Externe promotor: M. Breugelmans

drs ir.-arch. R. Verstraeten

2008 - 2009



artesis

industriële wetenschappen

Masterproef:

Virtueel ontwerpen met IFC Automatiseren meetstaat en kostprijsberekening

Kandidaat: Tim Lemoine

Promotoren

Interne promotor: ing. C. Vuye

Externe promotor: M. Breugelmans

drs ir.-arch. R. Verstraeten

2008 - 2009



artesis

industriële wetenschappen

Voorwoord

Deze thesis kwam tot stand in het kader van de Master opleiding van het departement Industriële Wetenschappen & technologie, Bouwkunde aan de Artesis Hogeschool Antwerpen.

Ik heb dit onderwerp voor verschillende redenen gekozen. Eerst en vooral behandelt deze thesis een nieuw thema, vooral in België, dat de bouwwereld drastisch kan en zal veranderen. Mijn passie voor informatica was de tweede reden voor deze keuze.

In deze thesis zal een eerste aanzet gegeven worden naar het automatisch genereren van een meetstaat. De thesis zal enkele eenvoudige gevallen behandelen. Er is zoveel mogelijk bij het programmeren getracht functies te schrijven die in de toekomst gemakkelijk uit te breiden zijn.

“De auteur geeft de toelating dit afstudeerwerk voor consultatie beschikbaar te stellen en delen van het afstudeerwerk te kopiëren voor eigen gebruik. Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit dit afstudeerwerk.”

4 juni 2009

Dankwoord

Ik wil in de eerste plaats mijn externe promotoren, Marc Breugelmans en Ruben Verstraeten, van harte bedanken voor de zeer goede begeleiding en het altijd klaar voor mij staan wanneer ik vragen had. Ik wil Marc Breugelmans nog eens extra bedanken voor het mogelijk maken van de maandelijkse meetings en de vergadering met de mensen van SCIA Engineer.

Daarnaast wil ik mijn interne promotor, ing. Cedric Vuye, bedanken voor het mogelijk maken van deze thesis en steeds tot mijner beschikking te staan wanneer ik vragen had.

Ook wil ik mijn medestudenten, Philip Gielis en Yves Van Looy, bedanken voor het openstaan voor mijn bevinden over het thema virtueel bouwen.

Als laatste wil ik mijn familie bedanken. Dankzij hun steun en zorg was het mogelijk om deze thesis tot een goed einde te brengen.

Tim Lemoine

4 juni 2009

Naam: Lemoine, Tim

Titel:

**Virtueel ontwerpen met behulp van IFC:
Automatiseren meetstaat en kostprijsberekening**

Interne promotor: ing. C. Vuye

Externe promotor: Marc Breugelmans

Functie: CAD-coördinator

Firma: Bolckmans NV

Industrieweg 4 2320 Hoogstraten

tel.: 03/340.23.24 **fax:** 03/314.45.59

e-mail: mbr@bolckmans.be

ir. Arch. Ruben Verstraeten

Functie: architect

Firma: Universiteit Gent

Jozef Plateaustraet 22 9000 Gent

tel.: 09/264.89.17 **fax:**

e-mail: ruben.verstraeten@ugent.be

Trefwoorden : BIM, virtueel bouwen, IFC, meetstaat, C#, Revit, buildingSMART

ABSTRACT

Vandaag de dag is het gebruik van computers in de bouwwereld niet meer weg te denken. Gedurende het ganse bouwproces wordt er gebruik gemaakt van IT-toepassingen. Eind jaren 1980 maakte de traditionele tekentafel plaats voor "Computer Aided Architectural Design", of kortweg CAAD. Door de jaren evolueerden de tekenapplicaties van 2D over 3D tot digitale bouwmodellen. Zulke digitale bouwmodellen worden ook wel eens Building Information Models genoemd.

Digitale bouwmodellen bevatten een grote hoeveelheid aan informatie waarmee het bouwproces doeltreffender kan verlopen. Door al deze informatie te koppelen, kunnen er heldere afspraken gemaakt worden. Om de één op één relatie tussen verschillende softwareapplicaties te optimaliseren is er het open bron formaat IFC in het leven geroepen.

Deze thesis onderzoekt daarom de mogelijkheden van BIM en IFC voor de automatisatie van een meetstaat en kostprijsberekening. Hiervoor wordt een eenvoudig programma ontwikkeld.

Nom: Lemoine, Tim

Titre:

**La construction virtuelle avec IFC:
Automatiser le métré en le calcul du coût**

Directeur de thèse interne: ing. C. Vuye

Directeur de thèse externe: Marc Breugelmans

Fonction: CAD-coördonateur

Firme: Bolckmans NV

Industrieweg 4 2320 Hoogstraten

tel.: 03/340.23.24 **fax:** 03/314.45.59

e-mail: mbr@bolckmans.be

ir. Arch. Ruben Verstraeten

Fonction: architecte

Firme: Universiteit Gent

Jozef Plateastraat 22 9000 Gent

tel.: 09/264.89.17 **fax:**

e-mail: ruben.verstraeten@ugent.be

Mots clef : BIM, Construction virtuelle, IFC, métré, C#, Revit, buildingSMART

RESUME DE LA THESE

Aujourd'hui, l'utilisation de l'ordinateur dans le milieu de la construction est devenue indispensable. Durant toute la procédure de la construction, nous utilisons les applications IT. Fin des années 80, la traditionnelle table de dessin a disparu pour laisser la place au « computer aided architectural design » appelé plus simplement « CAAD ». D'années en années, les applications des dessins ont évoluées de 2D à 3D pour arriver aux modèles de constructions digitaux. Certains modèles de construction digitaux sont parfois appelés « Building Information Models ».

Les modèles de constructions digitaux proposent une importante quantité d'informations permettant un travail plus performant. En regroupant toutes les informations, les renseignements sont plus clairs et précis. Le système IFC a été créé afin de permettre le passage d'un système à un autre sans perdre aucune donnée.

C'est pourquoi cette thèse recherche les possibilités de BIM et IFC afin d'automatiser le métré et le calcul du coût. C'est pourquoi un programme simple est en développement.

Name: Lemoine, Tim

Title:

**Virtual Building with IFC:
Automate the measurement and the costing**

Internal promotor: ing. C. Vuye

External promotor: Marc Breugelmans

Function: CAD-coordinator

Firm: Bolckmans NV

Industrieweg 4 2320 Hoogstraten

tel.: 03/340.23.24 **fax:** 03/314.45.59

e-mail: mbr@bolckmans.be

ir. Arch. Ruben Verstraeten

Function: architect

Firm: Universiteit Gent

Jozef Plateastraat 22 9000 Gent

tel.: 09/264.89.17 **fax:**

e-mail: ruben.verstraeten@ugent.be

Keywords : BIM, virtual building, IFC, measurement, C#, Revit

ABSTRACT

Today, the use of computers in the building industry is well established. During the entire process computer applications are used. At the end of 1980, the traditional drawing board made room for "Computer Aided Architectural Design", or simply CAAD. Over the years, drawing software evolved from 2D to 3D to digital building models. Such digital building models are also called Building Information Models.

Digital building models contain a large amount of information. By linking all this information, arrangements can be made more clearly. The open source format IFC was developed to ensure an one-to-one relationship between different software applications.

This thesis therefore examines the potential of BIM and IFC for the automation of measurement and costing. So a very simple application is developed.

Inhoudsopgave

Voorwoord	I
Dankwoord	II
Inhoudsopgave	VI
Lijst van Figuren	IX
Lijst van Codeblokken	XI
Lijst van Afkortingen	XIII
1. Inleiding & Probleemstelling	1
1.1. Inleiding.....	1
1.2. Probleemstelling	1
1.3. Doelstelling.....	2
2. Historisch overzicht	4
2.1. 2D ontwerpen	4
2.2. 3D ontwerpen	5
3. Building Information Modeling	7
3.1. Wat is BIM?	7
3.2. Geschiedenis	8
3.3. BIM methode	9
3.4. buildingSMART	9
3.4.1 IFC.....	10
3.4.2 IFD.....	10
3.4.3 IDM	11
4. Industry Foundation Classes	13
4.1. ISO 10303	13
4.1.1 Geschiedenis	13
4.1.2 Opbouw	14
4.2. Beschrijvingsmethode – EXPRESS	14
4.2.1 EXPRESS bouwstenen	16
4.2.1.1. Datatypes	16
4.2.1.2. Entiteit Attribuut	18
4.2.1.3. Supertypes en Subtypes	18

4.2.1.4. Algoritmische beperkingen	19
4.3. Implementatie methodes	19
4.3.1 STEP Physical File.....	19
4.3.2 STEP-XML.....	22
4.4. De structuur van IFC2x	24
4.4.1 De algemene architectuur.....	24
4.4.2 De Klassen	26
4.5. Ifc Roadmap	28
4.6. Certificering.....	28
4.7. Nieuwigheden in IFC2x4	30
5. Analyse van het informatiemodel.....	31
5.1. Opbouw van het Model	31
5.1.1 Ifcproject	31
5.1.2 IfcSite	32
5.1.3 IfcBuilding.....	33
5.1.4 IfcBuildingStorey	34
5.1.5 IfcSpace	34
5.2. Geometrische voorstelling.....	35
5.2.1 Algemeen.....	35
5.2.1.1. BoundingBox	35
5.2.1.2. SurfaceModel	36
5.2.1.3. Brep	37
5.2.1.4. Mappedrepresentation	38
5.2.2 Muren.....	38
5.2.3 Vloeren	41
5.2.4 Balken	43
5.2.5 Vensters en Deuren.....	45
5.3. Materialen.....	45
5.4. Plaatsing van objecten	48
6. Implementatie	50
6.1. Exporteren naar IFC	50
6.2. Inlezen van IFC-bestand	50

6.2.1 Functie OpenModel.....	52
6.2.2 Functie GetEntity.....	52
6.2.3 Functie GetEntities	53
6.2.4 Functie Version.....	53
6.3. Omzetting van klassen	53
6.4. Meeteenheden.....	54
6.5. Geometrie	54
6.6. Uitlezing van gegevens.....	55
6.6.1 Excel.....	55
6.6.2 PDF.....	56
6.7. Grafische User Interface	57
6.7.1 Project-eigenschappen.....	58
6.7.2 Viewer	60
6.7.3 Meetstaat	60
7. Tests	62
7.1. TestCase 1	62
7.1.1 Beschrijving	62
7.1.2 Probleemstelling.....	63
7.1.3 Uitwerking	64
7.2. Testcase 2.....	67
7.2.1 Beschrijving	67
7.2.2 Probleemstelling.....	69
7.2.3 Uitwerking	69
8. Besluit	72
Bibliografie	74

Lijst van Figuren

Figuur 1: Schematische opbouw van de applicatie.....	2
Figuur 2: Weergave traditioneel pakket (1)	5
Figuur 3: Centrale rol van BIM (13)	7
Figuur 4: BIM-pakketten die gebruikt of geëvalueerd worden onder de abonnees van AECbytes.....	8
Figuur 5: Weergave BIM methode (1).....	9
Figuur 6: Logo van buildingSMART (3)	10
Figuur 7: Logo van IFD Library (4).....	10
Figuur 8: Weergave van IFC/IFD/IDM (5)	12
Figuur 9: EXPRESS-G diagram voor het schema Familie	15
Figuur 10: Voorstelling van een datatype in EXPRESS-G.....	16
Figuur 11: Voorstelling van een entiteit en een abstracte entiteit.....	16
Figuur 12: Een opsommingstype in EXPRESS-G	16
Figuur 13: een gedefinieerd type in EXPRESS-G.....	17
Figuur 14: Een selectie type in EXPRESS-G.....	17
Figuur 15: Voorstelling van een attribuut en een optioneel attribuut	18
Figuur 16: Voorstellingen van de relatie tussen een supertype en zijn subtype.....	19
Figuur 17: De algemene architectuur van IFC en een UML-model voorbeeld van IfcWall en IfcSpace. (1)	25
Figuur 18: De IfcRoot klasse	26
Figuur 19: Fundamentele klassen	26
Figuur 20: De klasse IfcObject en zijn subtypes	27
Figuur 21: De klasse IfcRelationship en zijn subtypes.....	28
Figuur 22: De roadmap van IFC	28
Figuur 23: Certificering Logo's (8)	29
Figuur 24: De klasse IfcProject	31
Figuur 25: Ruimtelijke Structuur	32
Figuur 26: De klasse IfcSite.....	32
Figuur 27: De klasse IfcBuilding.....	33
Figuur 28: De klasse IfcBuildingStorey	34
Figuur 29: De klasse IfcSpace	34
Figuur 30: Bounding Box (9)	35
Figuur 31: Definitie van een IfcBoundingBox (9)	36
Figuur 32: SurfaceModel met Bounding Box (9)	37
Figuur 33: Brep met Bounding Box (9)	38
Figuur 34: De voorstelling van de as van een rechte en een gebogen muur (9)	39
Figuur 35: SweptSolid van een rechte en gebogen wand.....	40
Figuur 36: Clipping van een rechte en een gebogen wand (9)	41
Figuur 37: SweptSolid van een vloer (9).....	42

Figuur 38: Clipping van een vloer (9)	43
Figuur 39: SweptSolid van een balk (9)	44
Figuur 40: Clipping van een balk (9)	45
Figuur 41: IfcMaterialLayerSet (9).....	46
Figuur 42: IfcMaterialLayerSetUsage voor een muur (9).....	47
Figuur 43: IfcMaterialLayerSetUsage voor een vloer (9)	47
Figuur 44: IfcMaterialLayerSetUsage voor een schuin dak.....	48
Figuur 45: IfcObjectPlacement	48
Figuur 46: De geometrische hiërarchie via IfcLocalPlacement (1).....	49
Figuur 47: Exporteren van een model naar IFC in Autodesk Revit	50
Figuur 48: Omzetting afronding naar polygoon	55
Figuur 49: Voorstelling applicatie.....	57
Figuur 50: GUI van FLICT	58
Figuur 51: Venster met de eenheden	59
Figuur 52: Venster met de materiaallagen	59
Figuur 53: Weergave boomstructuur(links) en informatie van een element (rechts)	60
Figuur 54: Stuklijst	60
Figuur 55: Meetstaat	61
Figuur 56: 3D weergave van brandwand	62
Figuur 57: Vooraanzicht brandwand.....	62
Figuur 58: Schedule in Revit	63
Figuur 59: Export van een schedule in Revit	63
Figuur 60: Weergave van IfcTable.....	65
Figuur 61: Weergave van IfcTableRow.....	65
Figuur 62: Resultaat in FLICT	66
Figuur 63: Keuzevenster.....	66
Figuur 64: Weergave tabel Cellenbetonpaneel in Excel	67
Figuur 65: 3D weergave woning.....	68
Figuur 66: Weergaves van grondplan van gelijkvloers (links) en eerste verdieping (rechts) ..	68
Figuur 67: Vooraanzicht woning	69
Figuur 68: Meetstaat in FLICT.....	70
Figuur 69: Keuzelijst van ruimtelijke structuren	70
Figuur 70: Meetstaat in Excel	70

Lijst van Codeblokken

Code 1: EXPRESS weergave van het schema Familie	15
Code 2: Voorbeeld van een entiteit in IFC	16
Code 3: Opsommingstype in IFC	17
Code 4: Het gedefinieerd type IfcPositiveLengthMeasure in IFC	17
Code 5: Een selectie van types in IFC	17
Code 6: Voorstelling van een LIST in IFC	18
Code 7: Voorstelling van de entiteit IfcProject	19
Code 8: De header van een IFC bestand	20
Code 9: Externe mapping	21
Code 10: Interne mapping.....	21
Code 11: Het data-gedeelte van een IFC bestand.....	22
Code 12: Vergelijking tussen SPF en ifcXML	22
Code 13: Het iso_10303_28 element in ifcXML.....	22
Code 14: De header van een ifcXML bestand	22
Code 15: Het data-gedeelte van een ifcXML bestand.....	23
Code 16: Voorbeeld van het gebruik van een IFD Library voor een venster	30
Code 17: Gebruik van een bounding box in IFC	36
Code 18: Gebruik van een SurfaceModel in IFC.....	36
Code 19: Gebruik van Brep in IFC.....	37
Code 20: Curve2D voorstelling van een rechte muur	39
Code 21: SweptSolid van een rechte wand.....	40
Code 22: Clipping van een rechte wand.....	41
Code 23: Profile van een deur	45
Code 24: Opbouw van het XML bestand.....	51
Code 25: De gebruikte reguliere expressie	51
Code 26: Voorbeeldregel uit een IFC bestand	51
Code 27: Voorstelling van de implementatie van een Dictionary in C#	52
Code 28: De klasse Item	52
Code 29: Functie OpenModel	52
Code 30: Het gebruik van de functie OpenModel	52
Code 31: Functie GetEntity	53
Code 32: Gebruik van de functie GetEntity.....	53
Code 33: Functie GetEntities.....	53
Code 34: Gebruik van de functie GetEntities	53
Code 35: Functie Version	53
Code 36: Aanroepen van het project	53
Code 37: Omzetting naar SI eenheid.....	54
Code 38: Omzetting naar andere eenheid	54
Code 39: Omzetting naar andere eenheid in stringformaat	54

Code 40: Tabel in IFC.....	64
Code 41: Gebruikte LINQ query	66

Lijst van Afkortingen

BIM	Building Information Model
CAD	Computer Aided Design
DLL	Dynamically Linked Library
GUI	Grafische User Interface
IAI	International Alliance for Interoperability
IDM	Information Delivery Manual
IFC	Industry Foundation Classes
IFD	International Framework for Dictionaries
ISO	Internationale Organisatie voor Standaardisatie
LINQ	Language INtegrated Query
MEP	Mechanical, Electrical and Plumbing
ODBC	Open DataBase Connectivity
SDAI	Standard Data Access Interface
SPF	STEP Physical File
STEP	Standard for the Exchange of Product model data
SQL	Structured Query Language
XML	Extensible Markup Language

1. Inleiding & Probleemstelling

“Problems are only opportunities in work clothes.”

Henry John Kaiser, Industrialist

1.1. Inleiding

Vroeger was bouwen iets simpeler dan nu. Balken en stenen waren er in bepaalde maten en gewichten. Later namen de mogelijkheden toe en kreeg men meer en meer de behoefte om informatie op papier te zetten. De bouwtekeningen deden eind de negentiende eeuw hun intrede. Vervolgens werden de tekeningen bouwmodellen en werden er steeds meer eisen gesteld aan de informatie die de tekeningen moesten geven.

Vandaag de dag is het gebruik van computers in de bouwwereld daarom niet meer weg te denken. Gedurende het hele project, van ontwerp tot berekeningen, wordt er gebruik gemaakt van IT-toepassingen. Sinds de late jaren 1980 maakte de traditionele tekentafel plaats voor “Computer Aided Architectural Design”, of kortweg CAAD. Door de jaren evolueerden de tekenapplicaties van 2D over 3D tot digitale bouwmodellen. Zulke digitale bouwmodellen worden ook wel eens Building Information Models genoemd.

Digitale bouwmodellen bevatten een grote hoeveelheid aan informatie waarmee het bouwproces doeltreffender kan verlopen. Door al deze informatie te koppelen, kunnen er heldere afspraken gemaakt worden. Om de één op één relatie tussen verschillende softwareapplicaties te optimaliseren is er het open bron formaat IFC in het leven geroepen.

1.2. Probleemstelling

Elk jaar moet de bouw kosten maken om fouten te herstellen. Deze faalkosten treden onder andere op door problemen binnen het bouwproces, één van de belangrijkste oorzaken is de slechte samenwerking tussen de verschillende partijen. Fouten in het bouwproces nemen ook extra toe door een toenemende concurrentie en verhoogde tijdsdruk op de bouwmarkt. Door de faalkosten binnen het bedrijf te verlagen, kan de winst sterk stijgen. Elke euro die men bespaart, komt onmiddellijk ten goede van het bedrijf. Door het gebruik van BIM kunnen deze faalkosten verlaagd worden. Fouten in het begin van het bouwproces kunnen later voor grote problemen zorgen, zoals bijvoorbeeld een onvolledige en onnauwkeurige meetstaat.

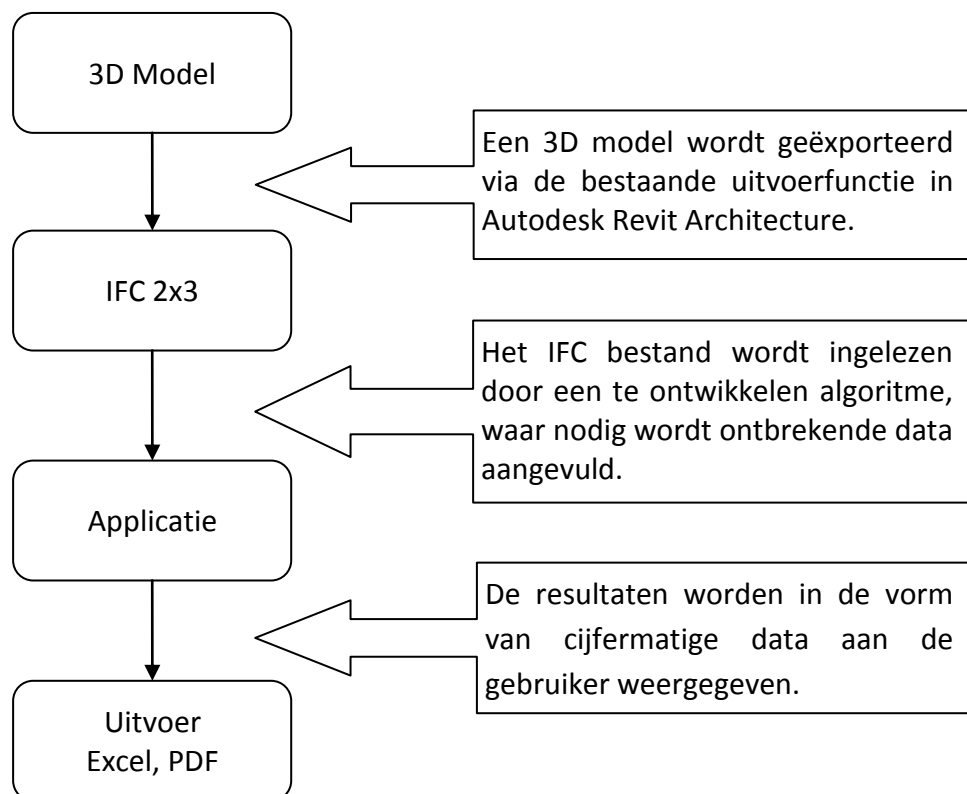
1.3. Doelstelling

Deze thesis onderzoekt daarom welke inspanningen er geleverd moeten worden om het opstellen van een meetstaat te verbeteren en te vereenvoudigen. Deze thesis bouwt verder op het onderzoek uit de thesis “Virtueel ontwerpen met IFC” van Kurt Schrauwen. Het hoofddoel van het onderzoek is:

Het automatisch genereren van een meetstaat en kostprijsberekening met behulp van Industry Foundation Classes (IFC).

De thesis bestaat uit twee grote delen. Deel 1 bestaat uit de literatuurstudie en zal handelen over Building Information Modelling en een uitgebreide uiteenzetting van de Industry Foundation Classes aan de hand van de ISO-norm 10303.

Deel 2 bevat de analyse van het model, de implementatie en de uitwerking van twee testcases. Beide testcases zullen in Autodesk Revit Architecture 2009¹ ontworpen worden en naar een IFC-formaat uitgevoerd worden. Voor het inlezen en verwerken dient een softwareapplicatie ontwikkeld te worden, voorzien van een overzichtelijke gebruikersinterface.



Figuur 1: Schematische opbouw van de applicatie

¹ <http://www.autodesk.nl/adsk/servlet/index?siteID=431528&id=12518232>

Deze thesis wordt mede begeleid door de firma Bolckmans NV en de Universiteit Gent.
Beide zijn zeer bedreven wanneer het aankomt op virtueel bouwen.

2. Historisch overzicht

“When we mean to build, We first survey the plot, then draw the model; And when we see the figure of the house, Then must we rate the cost of the erection; Which if we find outweighs ability, What do we then but draw anew the model.”

Shakespeare King Henry IV, 1597

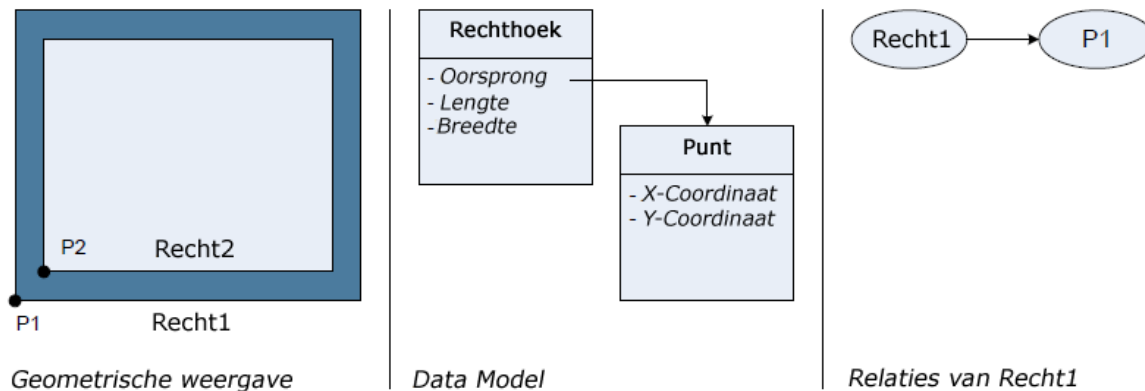
2.1. 2D ontwerpen

De geschiedenis van Computer Aided Design, of kortweg CAD is bijna zo oud als de geschiedenis van de computer. In het begin was er Sketchpad², een computerprogramma geschreven door Ivan Sutherland in de jaren 60 in het kader van zijn doctoraat. Sketchpad wordt beschouwd als de voorouder van de moderne CAD-toepassingen en als grote doorbraak in de ontwikkeling van computer graphics in het algemeen. De eerste commerciële toepassingen kwamen in de late jaren 60 en begin jaren 70 bij grote lucht- en ruimtevaart- en auto-bedrijven als General Motors en Lockheed. In de bouw duurde het meer dan 10 jaar voor CAD zijn ingang vond. Dit gebeurde met de komst van AutoCAD van Autodesk en ArchiCAD van Graphisoft in 1982 en 5 jaar later met de ontwikkeling van Microstation van Bentley Systems. Het gebruik van CAD-toepassingen werd bevorderd door de dalende prijzen van computers, waardoor CAD ook voor kleinere bedrijven betaalbaar werd.

Er was echter een groot verschil in het gebruik van CAD-toepassingen tussen de verschillende industrieën. In tegenstelling tot de bouw, was de mechanische industrie in het begin al geïnteresseerd in het gebruik van 3D-vormen. Het geometrische model van de meeste CAD-toepassingen in de bouwsector is gebaseerd op eenvoudige tweedimensionale vormen, zoals punten, lijnen, rechthoeken, polylines, wat neerkomt op muren, ramen, deuren, ...

Figuur 2 toont de weergave van een kamer gemodelleerd in een traditioneel tekenpakket.

² <http://en.wikipedia.org/wiki/Sketchpad>



Figuur 2: Weergave traditioneel pakket (1)

Het grote probleem bij deze manier van werken is dat het geen echte objecten zijn. De kamer bestaat niet uit 4 muren, maar uit 2 rechthoeken. Extra informatie moet aan de tekening toegevoegd worden aan de hand van tekst. Dit maakt deze tekeningen onbruikbaar voor andere doeleinden dan visualisatie.

2.2. 3D ontwerpen

Om de visualisatie te verbeteren werden er 3D-mogelijkheden aan de CAD-toepassingen toegevoegd. De eerste 3D-weergaves konden best omschreven worden als 2,5D. Deze weergaves waren gebaseerd op 2D ontwerpen waaraan een hoogte werd meegegeven. Dit was mogelijk doordat men werkte met horizontale vloeren en rechtopstaande wanden. Voor ingewikkelde constructies had men meer nodig dan de hoogte om een 3D-weergave te kunnen maken.

De huidige CAD-toepassingen maken het mogelijk om “echt” 3D te modelleren, afgeleid uit de mechanische industrie. Bij 3D ontwerpen krijgt men een visuele voorstelling van het project, er wordt ontworpen zoals men gaat bouwen. Hierdoor kunnen fouten sneller opgespoord worden.

Autodesk zette met de ontwikkeling van AutoCad Architectural Desktop³ een eerste stap naar het gebruik van bouwkundige elementen of objecten. Deze elementen zijn echter nog geen echte objecten, maar blijven een bundel van lijnen en punten. Door meer en meer informatie aan het 3D model te koppelen verkrijgt men een intelligent model. Zo komt men uiteindelijk tot het zogenaamde Building Information Model.

³ AutoCad Architectural Desktop, of ADT is de standaard AutoCad uitgebreid met bouwkundige objecten. Momenteel is ADT bekend onder de naam AutoCad Architecture.

BIM in de bouwsector kan vergeleken worden met de introductie van objectgeoriënteerde programmeertalen in de software-industrie. Maar door een gebrek aan standaardisatie blijven vele architecten en ingenieurs werken op de traditionele tweedimensionale manier.

3. Building Information Modeling

“BIM is not just software but a methodology of practice.”

Huw Robets, Bentley Systems

3.1. Wat is BIM?

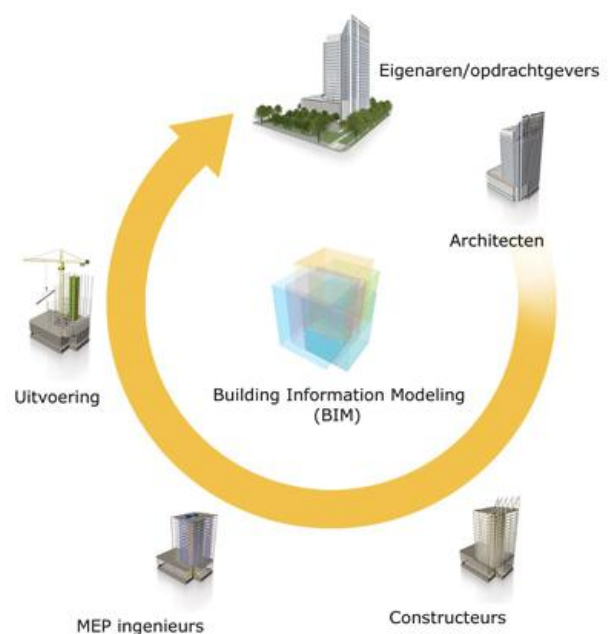
BIM staat voor Building Information Model of de bewerking ervan en dan spreken we over Building Information Modeling. Het maakt gebruik van driedimensionale, dynamische modelleringssoftware om de productiviteit tijdens het ontwerp en de bouw te verhogen. Het model omvat geometrie, geografische gegevens, hoeveelheden en eigenschappen van bouwelementen (bijvoorbeeld details van de fabrikant).

Alle relevante informatie wordt gedurende het hele bouwproces opgeslagen, gebruikt en beheerd in het digitale model. Hierdoor werken alle partijen die bij het bouwproces betrokken zijn met dezelfde informatie. Die informatie is altijd beschikbaar en up to date.

Het proces gaat veel verder dan de omschakeling naar nieuwe software. Het vereist ook aanpassingen naar de samenwerking en uitwisseling van gegevens tussen architecten en ingenieurs.

De bron aan informatie is dus wat het verschil maakt tussen een 3D CAD en een BIM. Het voegt een extra dimensie toe aan

het driedimensionale, we kunnen als het ware spreken over 3D+. Een BIM kan bestaan in een 3D, 4D of nD model, net hoever men wil gaan. Een goede BIM is een model waar alle nodige informatie inzit om een gebouw zo optimaal te kunnen bouwen en gebruiken. De informatie kan bestaan uit het materiaal tot de fysische eigenschappen. Zo kan bijvoorbeeld het gewicht worden bepaald, als er een soortelijke massa aan het element wordt toegevoegd. En kan er voor de start van de bouwwerken al bepaald worden welk type van kraan nodig is om het element te hijsen. De grenzen tussen architect, ingenieurs, calculators en werkvoorbereiders zullen hierdoor vervagen. De traditionele tekenaar zal verdwijnen en plaats maken voor een virtueel bouwer. Er kan dan beroep gedaan worden op een zogenaamde BIM-manager. Deze persoon zorgt voor een goede coördinatie om gegevens over het project te verzamelen en uit te wisselen.



Figuur 3: Centrale rol van BIM (13)

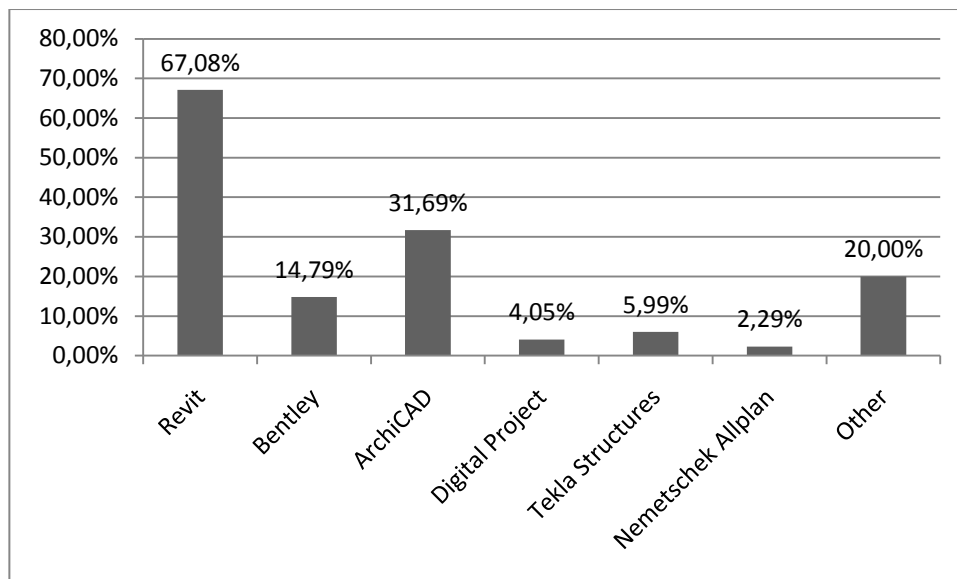
Naast BIM zijn er nog andere Informatie Modellen, zo kennen we ook Facility Information Model (FIM) en Geo Information Model (GIM). Een FIM regelt de faciliteiten (elektriciteit, verwarming, water) in een gebouw. FIM is een logische opvolging van een BIM, maar kan er ook losgemaakt van worden. Een GIM bevat informatiemodellen op stedenbouwkundige schaal. Zo kan er snel een overzicht van functies gemaakt worden in een bepaald gebied.

3.2. Geschiedenis

Algemeen wordt aangenomen dat professor Charles M. Eastman van het Georgia Institute of Technology de term bedacht heeft, in veronderstelling dat de term Building Information Model hetzelfde betekent als Building Product Model, die hij gebruikt heeft in zijn boek⁴ en papers sinds de late jaren 1970.

De eerste uitvoering van BIM was het Virtual Building-concept door Graphisoft's Archicad in haar debuut in 1987. Een andere applicatie gebaseerd op BIM werd ontwikkeld door Revit Technology Corporation. In 2002 werd dit bedrijf door Autodesk opgekocht.

Vandaag de dag zijn er een heleboel applicaties op de markt die de BIM-methode implementeren. Uit een peiling in 2007 (2) door AECbytes blijkt dat Revit van Autodesk de meest gebruikte software is, andere grote spelers zijn ArchiCAD van Graphisoft en Microstation van Bentley Systems.



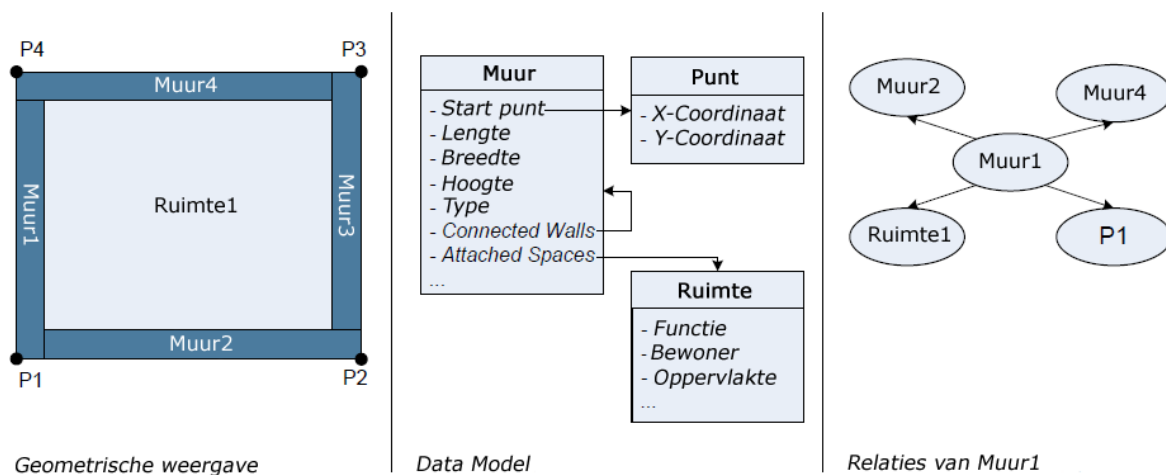
Figuur 4: BIM-pakketten die gebruikt of geëvalueerd worden onder de abonnees van AECbytes

⁴ Building Product Models: Computer Environments Supporting Design and Construction. 1999, Boca Raton, FL: CRC Press

3.3. BIM methode

De informatie van het BIM-proces wordt in een centrale databank opgeslagen, het BIM-model. Elk object heeft een eigen betekenis en beschrijving. Dit houdt in dat er nu ontworpen kan worden met muren, vloeren, daken, enz. in plaats van met lijnen en rechthoeken om dezelfde objecten weer te geven.

Volgende figuur visualiseert de BIM idee zeer kort aan de hand van dezelfde voorstelling als in Figuur 2. In plaats van twee rechthoeken wordt de kamer nu bepaald door vier muren en een ruimte. De muur heeft een start punt, een lengte, een hoogte en een breedte, maar bevat nog meer informatie over de aangesloten muren, de ruimte, het type en eventueel andere attributen die van belang zijn voor dit object. Deze keer is de muur bewust van het feit dat het een muur is, net zoals hij zich bewust is van zijn omgeving. Dit maakt het mogelijk deze informatie in verschillende contexten te gebruiken.



Figuur 5: Weergave BIM methode (1)

3.4. buildingSMART

In 1994 werd het International Alliance for Interoperability⁵ opgericht met de doelstelling een open en softwareonafhankelijke uitwisselingsstandaard te ontwikkelen voor de bouwsector. Om dit te bereiken werd er een internationale samenwerking opgestart van tientallen bouwkundige ingenieurs om een technische oplossing te creëren. Deze samenwerking heeft geresulteerd in het open uitwisselformaat IFC, de IFD bibliotheek en de IDM Learning Guide.

⁵ <http://www.buildingsmart.com/>



Figuur 6: Logo van buildingSMART (3)

3.4.1 IFC

Softwareapplicaties slaan het Informatie Model meestal op in een eigen bestandsformaat, zoals bijvoorbeeld .rvt in Autodesk Revit. Om deze waardevolle informatie beschikbaar te maken voor andere applicaties, kunnen deze applicaties bestandsformaten van andere softwarefabrikanten ondersteunen of kunnen ze IFC gebruiken voor deze uitwisseling. IFC gebruikt ISO STEP⁶ als primaire technologie. De structuur is beschreven in de taal EXPRESS. De gegevens kunnen op twee manieren worden vastgelegd, text-based of server-based. In text-based onderscheiden we twee bestandsformaten, SPFF en ifcXML. SPFF staat voor STEP Physical File Format of korter SPF en is momenteel het meest gebruikte bestandsformaat voor de uitwisseling van een BIM. De andere, modernere manier van opslag maakt gebruik van XML technologieën. Het schema wordt beschreven in de eXtensible Schema Definition (XSD) taal en de bijhorende data in eXtensible Markup Language (XML).

Een nieuwe manier van opslag is server-based. Het data model wordt nu niet meer weggeschreven naar een bestand, maar rechtstreeks in een databank opgeslagen. Het voordeel van deze manier van werken is dat het volledige model op een plaats wordt bewaard en altijd up-to-date is. Gebruikers kunnen dan inloggen op de server en de benodigde informatie uit het model opvragen. Er moet niet meer geknoeid worden met het doorsturen van bestanden. Een voorbeeld van zo'n server is de BIMserver⁷.

In hoofdstuk 0 zal er verder ingegaan worden op de technische aspecten van IFC.

3.4.2 IFD

IFD staat voor "International Framework for Dictionaries". De naam wordt zowel gebruikt voor de bibliotheken als voor de organisatie die instaat voor de instandhouding ervan.



Figuur 7: Logo van IFD Library (4)

⁶ [http://www.tc184-sc4.org/SC4_Open/SC4_Legacy_Products_\(2001-08\)/STEP_\(10303\)/](http://www.tc184-sc4.org/SC4_Open/SC4_Legacy_Products_(2001-08)/STEP_(10303)/)

⁷ De BIMserver is een Nederlands project van TNO en de TU Eindhoven. <http://www.bimserver.org>

IFD is in eenvoudige termen, een norm voor terminologie bibliotheken of ontologieën. De IFD Library is naast IFC en IDM een van de belangrijkste onderdelen van de buildingSMART technologie. Het concept voor de IFD Library is afgeleid van de internationaal aanvaarde open standaarden die zijn ontwikkeld door de ISO (belangrijkste ISO 12006-3:2007). IFD is een naslagbibliotheek bestemd voor de ondersteuning van verbeterde interoperabiliteit in de bouwsector. Het biedt een flexibele methode voor het koppelen van bestaande databanken met bouw informatie aan een BIM. Vanwege de meertalige mogelijkheden van de IFD Library is deze koppeling taalonafhankelijk.

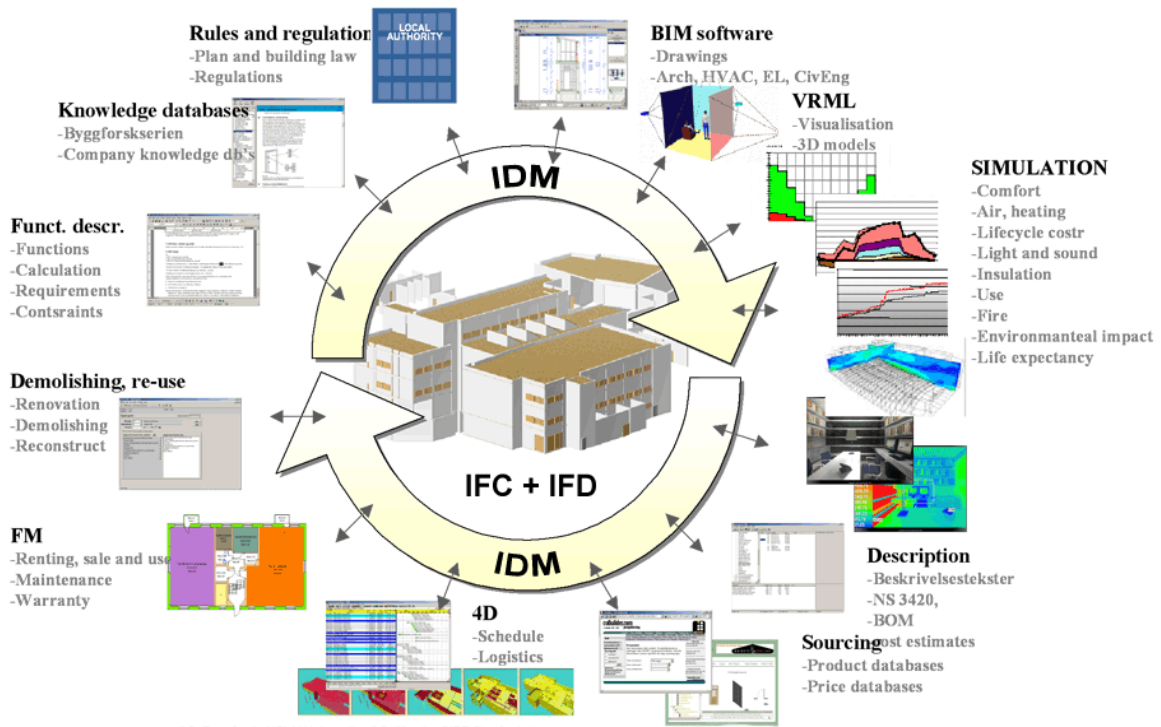
IFD is in 2006 ontstaan uit de samenwerking tussen het BARBI-project van de Noorse tak van buildingSMART⁸ en het LexiCon-project van STABU⁹ in Nederland. IFD zal volledig ondersteund worden in de volgende release van IFC2x, IFC2x4.

3.4.3 IDM

Men is zich er van bewust geworden dat bij de communicaties tussen bouwpartners niet alleen afspraken over informatie overeengekomen moeten worden, maar ook afspraken gemaakt moeten worden in verband met processen. Met die intentie is de Information Delivery Manual (IDM) in het leven geroepen. Het doel is om een basis te verschaffen voor betrouwbare informatie, zodat de gebruiker erop kan vertrouwen dat de informatie die zij ontvangt juist en voldoende is voor de activiteiten die hij moet uitvoeren. De basismethoden van IDM voor het definiëren van bedrijfsprocessen en het opgeven van de informatie-uitwisseling zijn onafhankelijk van eender welk informatie model. IDM biedt ook technische oplossingen aan die gebruik maken van specifieke informatie modellen. Voor de bouwsector maakt het gebruik van IFC en de eigenschappen die gedefinieerd zijn in de IFD bibliotheek. Voor een overzicht van IFC/IDM/IFD zie Figuur 8.

⁸ <http://buildingsmart.no/>

⁹ <http://www.stabu.org/>



Figuur 8: Weergave van IFC/IFD/IDM (5)

4. Industry Foundation Classes

“Compatible: Gracefully accepts erroneous data from any source.”

Onbekend

In dit hoofdstuk wordt er dieper ingegaan op het data-model van IFC en wordt er uitgelegd hoe IFC in staat is de uitwisseling van gegevens tussen verschillende applicaties tot stand te brengen. Daarnaast wordt er ook al gekeken naar de nieuwe editie van IFC.

4.1. ISO 10303

ISO-10303 is een ISO-norm voor de weergave en de uitwisseling van product gegevens. De officiële titel is “Industrial automation systems and integration – Product data representation and exchange” of beter bekend als “Standaard voor de uitwisseling van Product model data”, afgekort tot STEP.

Het doel van deze ISO-norm is een techniek te voorzien voor de beschrijving van product data gedurende de levenscyclus van een product, onafhankelijk van het systeem. De aard van deze beschrijving maakt het niet alleen geschikt van een neutrale manier van bestandsuitwisseling, maar vormt ook de basis voor de uitvoering en het delen van productdatabases en archivering.

STEP wordt ontwikkeld en onderhouden door de technische commissie ISO TC 184¹⁰, Technische Industriële automatisering en integratie, subcomité SC4, Industriële gegevens.

4.1.1 Geschiedenis

De ontwikkeling van STEP kan onderverdeeld worden in drie fasen. De ontwikkeling van STEP begon in 1984 als opvolger van IGES¹¹, SET en VDA-FS¹². De oorspronkelijke idee was om STEP te baseren op een enkel, maar volledig implementatieonafhankelijk Product Informatie Model. Maar vanwege de grote complexiteit werd de norm opgedeeld in kleinere delen, die afzonderlijk ontwikkeld en goedgekeurd konden worden. In 1994 werd de oorspronkelijke versie van STEP als ISO-norm gepubliceerd met de delen 1, 11, 21, 31, 41, 42, 43, 44, 46, 101, AP201 and AP203. Vandaag is deel AP203 nog steeds het belangrijkste onderdeel van STEP en wordt door verschillende CAD-toepassingen ondersteund voor hun import en export functies.

In een tweede fase werd STEP door toedoen van de ruimtevaart, automobiëlindustrie en elektronische industrie sterk uitgebreid. Deze fase werd in 2002 met een tweede grote release gepubliceerd.

¹⁰ <http://www.tc184-sc4.org/>

¹¹ Het Initial Graphics Exchange Specification definieert een neutraal gegevensformaat waarmee de uitwisseling van informatie tussen CAD-systemen.

¹² VDA-FS is een CAD-formaat voor uitwisseling van gegevens voor de overdracht van het ene systeem naar het andere.

De eerste en tweede release waren te groot, overlaptten met elkaar en waren niet voldoende geharmoniseerd. Deze tekortkomingen leidden tot de ontwikkeling van de STEP modulaire architectuur (delen 400 en 1000).

4.1.2 Opbouw

STEP is in vele verschillende delen opgedeeld en gegroepeerd in:

- Omgeving
 - Delen 1x: Beschrijvingsmethoden: EXPRESS, EXPRESS-X
 - Delen 2x: Uitvoering: STEP-File, STEP-XML, SDAI
 - Delen 3x: conformiteitsniveau, testmethodologie en framework
- Geïntegreerde datamodellen
 - Het integrale Resources (IR), bestaande uit
 - Delen 4x en 5x: Geïntegreerde generieke middelen
 - Delen 1xx: Geïntegreerde toepassingsmiddelen
 - PLIB ISO 13584-20 *Delen bibliotheek: Logisch model van expressies*
 - Delen 5xx: Toepassing geïntegreerde constructies (AIC)
 - Delen 1xxx: Application Modules (AM)
- Bovenste delen
 - Delen 2xx: Application Protocols (AP)
 - Delen 3xx: Abstract Test Suites (ATS) voor AP
 - Delen 4xx: Implementatie modules voor AP

In totaal bestaat STEP uit honderden delen en worden er jaarlijks nieuwe delen toegevoegd of herzieningen van oudere delen vrijgegeven. Dit maakt dat STEP de grootste standaard is binnen ISO.

De delen 1x en 2x zijn voor deze thesis van toepassing en zullen hieronder verder besproken worden.

4.2. Beschrijvingsmethode – EXPRESS

IFC wordt beschreven door EXPRESS. EXPRESS is een data modelleertaal voor product gegevens en is geformaliseerd in de ISO-norm 10303 (STEP) en gestandaardiseerd als ISO 10303-11 (6).

Een EXPRESS model kan op twee manieren worden gedefinieerd: tekstueel of grafisch. Voor verificatie en als input voor tools zoals SDAI¹³ is de tekstuele voorstelling in een ASCII¹⁴-bestand het meest belangrijk. De grafische voorstelling, genaamd EXPRESS-G is beter geschikt voor menselijk gebruik. In EXPRESS-G kunnen niet alle details weergegeven worden zoals die beschreven staan in de tekstuele vorm. EXPRESS is vergelijkbaar met de programmeertaal PASCAL. Binnen een schema kunnen verschillende datatypes gedefinieerd worden, samen met de structurele beperkingen en algoritmische regels.

Een eenvoudig data model ziet er als volgt uit:

```

SCHEMA Familie;

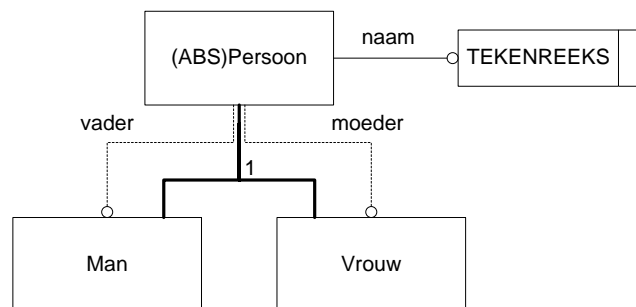
ENTITY Persoon
  ABSTRACT SUPERTYPE OF (ONEOF (Man, Vrouw));
  naam: STRING;
  moeder: OPTIONAL Vrouw;
  vader: OPTIONAL Man;
END_ENTITY;

ENTITY Vrouw
  SUBTYPE OF (Persoon);
END_ENTITY;

ENTITY Man
  SUBTYPE OF (Persoon);
END_ENTITY;

END_SCHEMA;
    
```

Code 1: EXPRESS weergave van het schema Familie



Figuur 9: EXPRESS-G diagram voor het schema Familie

Het data model is ingesloten in het schema *Familie*. Het bevat een abstract supertype entiteit *Persoon* met twee subtypes *Man* en *Vrouw*. Door het opleggen van abstractie kan maar een van beide (ONEOF) voorvallen. Elk optreden van een persoon heeft een verplichte attribuut *naam* en eventueel attributen *vader* en *moeder*. Er is een vaste stijl voor het lezen van de kenmerken van enkele entiteiten:

¹³ De term SDAI staat voor Standard Data Access Interface en wordt gebruikt voor applicaties die STEP ondersteunen

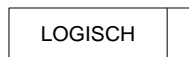
¹⁴ ASCII is een afkorting van American Standard Code for Information Interchange en is een standaard om een aantal tekens weer te geven en aan ieder teken een geheel getal te koppelen.

- Een vrouw kan de rol spelen van de moeder van een persoon
- Een man kan de rol spelen van de vader van een persoon

4.2.1 EXPRESS bouwstenen

4.2.1.1. Datatypes

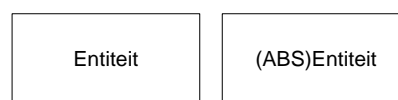
- Eenvoudige datatypes:
 - REAL: decimale getallen. Het decimaal punt moet altijd aanwezig zijn, zelfs als het getal een geheel getal is
 - INTEGER: een geheel getal
 - NUMBER: een getal dat zowel real als integer kan zijn
 - LOGICAL: heeft de waarden true, false en unknown
 - BOOLEAN: heeft de waarden true of false
 - BINARY: een opeenvolging van bits, die elk de waarde 0 of 1 kunnen hebben
 - STRING: een reeks van tekens



Figuur 10: Voorstelling van een datatype in EXPRESS-G

- Entiteitstype

Dit is het belangrijkste datatype binnen EXPRESS. Het kan op twee manieren gerelateerd worden: in een sub-supertype boom en/of attributen.



Figuur 11: Voorstelling van een entiteit en een abstracte entiteit

```

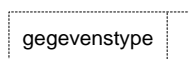
ENTITY IfcApprovalRelationship;
  RelatedApproval : IfcApproval;
  RelatingApproval : IfcApproval;  Description : OPTIONAL IfcText;
  Name : IfcLabel;
END_ENTITY;

```

Code 2: Voorbeeld van een entiteit in IFC

- Opsommingstype

De waarden van een opsomming zijn eenvoudige tekenreeksen, zoals rood, groen en blauw voor een RGB-opsomming.



Figuur 12: Een opsommingstype in EXPRESS-G


```

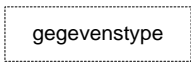
TYPE IfcWallTypeEnum = ENUMERATION OF
  (STANDARD
  , POLYGONAL
  , SHEAR
  , ELEMENTEDWALL
  , PLUMBINGWALL
  , USERDEFINED
  , NOTDEFINED) ;
END_TYPE ;

```

Code 3: Opsommingstype in IFC

- Gedefinieerd type

Ze kunnen gebruikt worden om andere datatypes verder te specialiseren.



Figuur 13: een gedefinieerd type in EXPRESS-G

```

TYPE IfcPositiveLengthMeasure = IfcLengthMeasure;
WHERE
  WR1 : SELF > 0.;
END_TYPE ;

```

Code 4: Het gedefinieerd type IfcPositiveLengthMeasure in IFC

- Selectie type

Selecteert een keuze tussen verschillende opties. De meest gebruikte selecties zijn diegene tussen entiteiten.



Figuur 14: Een selectie type in EXPRESS-G

```

TYPE IfcMaterialSelect = SELECT
  (IfcMaterial,
  , IfcMaterialList
  , IfcMaterialLayerSetUsage
  , IfcMaterialLayerSet
  , IfcMaterialLayer) ;
END_TYPE ;

```

Code 5: Een selectie van types in IFC

- Aggregation type

- ARRAY: is een geordende verzameling met een vaste grootte en wordt weergegeven als A[1:?]

- BAG: is een ongeordende verzameling die dubbelen mag bevatten en wordt weergegeven als B[1:?]
- LIST: is een geordende verzameling en wordt weergegeven als L[1:?]
- SET: is een ongeordende verzameling die geen dubbelen mag bevatten en wordt weergegeven als S[1:?]

Binnen IFC worden LIST en SET het meest gebruikt.

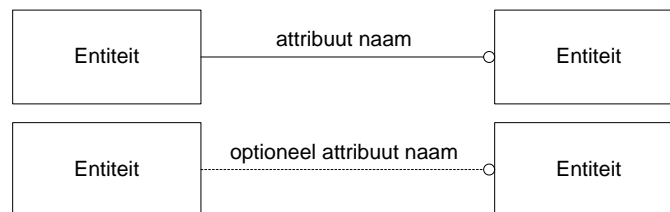
```
ImplGeoTopStartHeights : LIST [1:?] OF IfcLengthMeasure;
```

Code 6: Voorstelling van een LIST in IFC

4.2.1.2. Entiteit Attribuut

Er zijn drie verschillende soorten attributen, expliciete, afgeleide en inverse. Al deze kunnen opnieuw gedeclareerd worden in een subtype.

- Expliciete attributen zijn rechtstreeks zichtbaar
- Afgeleide attributen krijgen hun waarde via een expressie. Deze expressie verwijst in de meeste gevallen naar andere attributen, maar kan ook gebruik maken van functies
- Inverse attributen voegen geen extra “informatie” toe aan een entiteit, maar leggen een verband met een andere entiteit

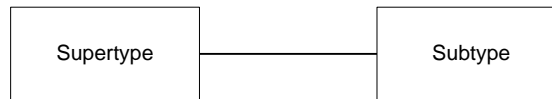


Figuur 15: Voorstelling van een attribuut en een optioneel attribuut

4.2.1.3. Supertypes en Subtypes

Een entiteit kan gedefinieerd worden als een subtype van één of meerdere entiteiten. Een supertype kan een willekeurig aantal subtypes bevatten. Het is in STEP heel gebruikelijk zeer complexe sub-supertype structuren op te bouwen.

Een entiteit kan gebouwd worden als een enkele entiteit of als onderdeel van een complexe structuur. Om het aantal mogelijke combinaties in te perken, heeft men speciale beperkingen op niveau van het supertype ingevoerd, zoals ONEOF en TOTALOVER.



Figuur 16: Voorstellingen van de relatie tussen een supertype en zijn subtype

4.2.1.4. Algoritmische beperkingen

Entiteiten en gedefinieerde datatypes kunnen verder beperkt worden door WHERE-regels. Een WHERE-regel is een uitdrukking die moet nagaan of een EXPRESS schema geldig is of niet. Net als afgeleide attributen mogen de beperkingen zich beroepen op functies, die zich verder kunnen beroepen op procedures.

```

ENTITY IfcProject
  SUBTYPE OF (IfcObject);
  LongName : OPTIONAL IfcLabel;
  Phase : OPTIONAL IfcLabel;
  RepresentationContexts : SET [1:?] OF IfcRepresentationContext;
  UnitsInContext : IfcUnitAssignment;
  WHERE
    WR31 : EXISTS (SELF\IfcRoot.Name) ;
    WR32 : SIZEOF (QUERY (Temp <* RepresentationContexts |
      'IFC2X3.IFCGEOMETRICREPRESENTATIONSUBCONTEXT' IN TYPEOF (Temp)
    )) = 0;
    WR33 : SIZEOF (SELF\IfcObjectDefinition.Decomposes) = 0;
  END_ENTITY;
  
```

Code 7: Voorstelling van de entiteit *IfcProject*

Bovenstaande code beschrijft dat de entiteit *IfcProject* enkel kan bestaan wanneer het attribuut *naam* ingevuld is en de entiteit niet ontleed kan worden.

4.3. Implementatie methodes

4.3.1 STEP Physical File

SPF is de meest gebruikte vorm van uitwisseling van STEP-gegevens. Vanwege de ASCII-structuur is het eenvoudig te lezen met een typisch geval per regel. Een SPF wordt ook wel eens p21-bestand genoemd. Het bestand heeft meestal .stp of .step als uitgang maar voor het gebruik met Industry Foundation Classes krijgt het de uitgang .ifc mee.

Een SPF is opgedeeld in twee grote blokken, het header- en het data-gedeelte. De header (zie Code 8) heeft een vaste structuur bestaande uit 3 tot 6 groepen in een gegeven volgorde. Alle velden kunnen lege tekenreeksen bevatten, behalve de velden *time_stamp* en *FILE_SCHEMA*.

- **FILE_DESCRIPTION:**
 - Description: algemene beschrijving van het bestand
 - implementation_level: de versie en de conformiteit van dit bestand. Mogelijke versies zijn: 1 voor de oorspronkelijke norm van 1994, 2 voor de

technische herziening in 1995 en 3 voor de tweede editie van de technische herziening. Voor IFC wordt altijd 2;1 gebruikt.

- **FILE_NAME:**
 - Naam van de uitwisselingsstructuur. Het kan overeenkomen met de naam van het bestand.
 - `time_stamp`: geeft aan waarop het bestand gemaakt is. De tijdnotatie wordt weergegeven volgens de norm ISO 8601, bijvoorbeeld 2009-05-20T18:47:45
 - `auteur`: naam van de persoon die het bestand heeft gecreëerd.
 - Organisatie tot wie de auteur behoort
 - `preprocessor_version`: de naam en de versie van het systeem dat het bestand produceert
 - `originating_system`: de oorspronkelijke naam en versie van het systeem dat het bestand geproduceerd heeft
 - `authorisation`: de naam van de persoon die gemachtigd is voor dit bestand

- **FILE_SCHEMA:** geeft de IFC versie weer waarin het bestand is opgesteld, bv. IFC2X3

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(
/* description */      ('IFC2X_PLATFORM'),
/* implementation_level */ ('2;1');
FILE_NAME(
/* name */            'test_project',
/* time_stamp */      '2002-10-17T21:25:27+06:00',
/* author */          ('user name'),
/* organization */    ('user organization'),
/* preprocessor_version */ 'IFC toolbox name',
/* originating_system */ 'software application name',
/* authorisation */    'name of authority');
FILE_SCHEMA(
/* longform schema name */ ('IFC2X3'));
ENDSEC;
```

Code 8: De header van een IFC bestand

Het data-gedeelte (zie Code 11) bevat de gegevens volgens een specifiek EXPRESS schema. De codering van deze gegevens volgt een aantal zeer eenvoudige principes:

- **Instantienaam:** elke entiteit begint met een unieke naam “#1234”. De naam moet bestaan uit een positief getal en is meestal kleiner dan 2^{63} . De unieke naam wordt gebruikt om naar andere entiteiten te verwijzen.

- Een entiteit wordt weergegeven door de volledige naam van de entiteit in hoofdletters, gevolgd door de attributen tussen de haakjes. Bv. “#14=IFCDIRECTION(...)”
- Instanties van complexe entiteiten kunnen ofwel intern of extern gelinkt zijn, zogenaamde mapping:
 - Externe mapping: de instantienaam wordt opgegeven als attribuut

```
#3=IFCCARTESIANPOINT((0.,0.,0.));
#26=IFCAXIS2PLACEMENT3D(#3,$,$);
```

Code 9: Externe mapping

- Interne mapping: de entiteit wordt nu als attribuut ingevuld

```
#1223=IFCPROPERTYSSINGLEVALUE('Rafter Cut',$,IFCINTEGER(33615),$);
```

Code 10: Interne mapping

- Weergave van attributen:
 - Enkel expliciete attributen worden weergegeven. Inverse, afgeleide of opnieuw gedeclareerde attributen worden niet weergegeven, omdat deze attributen afgeleid kunnen worden uit de andere attributen.
 - Niet ingevulde attributen worden weergegeven door “\$”. Dit gebeurt vooral bij optionele attributen.
 - Expliciete attributen die opnieuw gedeclareerd worden in een subtype krijgen een “*” op de positie van het supertype
- Weergave van andere gegevens:
 - Opsomming, boolean en logische waarden worden in hoofdletters weergegeven beginnend en eindigend met een punt. Bv. “.TRUE.”
 - Stringwaarden worden tussen “” geplaatst.
 - De elementen van het type aggregation (ARRAY, BAG, SET, LIST) worden tussen haakjes weergegeven, gescheiden door “,”

```
DATA;
#1=IFCORGANIZATION($,'Revit 2008',$,$,$);
#2=IFCAPPLICATION(#1,'2008','Revit 2008','Revit');
#4=IFCCARTESIANPOINT((0.,0.));
#5=IFCDIRECTION((1.,0.,0.));
#10=IFCDIRECTION((0.,0.,-1.));
#11=IFCDIRECTION((1.,0.));
#12=IFCDIRECTION((-1.,0.));
#13=IFCDIRECTION((0.,1.));
#14=IFCDIRECTION((0.,-1.));
#15=IFCSIUNIT(*,.LENGTHUNIT.,.MILLI.,.METRE.);
#16=IFCSIUNIT(*,.AREAUNIT.,.MILLI.,.SQUARE_METRE.);
```

```

. . .
ENDSEC;
END-ISO-10303-21;

```

Code 11: Het data-gedeelte van een IFC bestand

4.3.2 STEP-XML

STEP-XML (7) maakt gebruik van de eXtensible Schema Definition (XSD) taal om een EXPRESS schema voor te stellen. Voor de weergave van de data wordt eXtensible Markup Language (XML) gebruikt. Het heeft dan ook de passende naam ifcXML meegekregen.

Hieronder wordt een simpel voorbeeld gegeven van de weergave van eenzelfde entiteit in SPF en ifcXML.

```

SPF:
    #84 = IFCPROPERTY SINGLEVALUE ('Red', $, IFCINTEGER(255), $);

ifcXML:
    <IfcPropertySingleValue id="i84">
        <Name>Red</Name>
        <NominalValue>
            <IfcInteger>255</IfcInteger>
        </NominalValue>
    </IfcPropertySingleValue>

```

Code 12: Vergelijking tussen SPF en ifcXML

De header van ifcXML bestaat uit twee delen, een header- en een data-gedeelte. Beide elementen zijn omhuld door het element “iso_10303_28”.

```

<iso_10303_28 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"
xsi:schemaLocation="urn:iso.org:standard:10303:part(28):version(2):xmlschema:
common http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/ex.xsd" version="2.0">

```

Code 13: Het iso_10303_28 element in ifcXML

De header, “iso_10303_28_header” bevat net zoals bij SPF informatie over de naam, auteur, datum, enz.

```

<iso_10303_28_header>
    <name>Case 1 </name>
    <time_stamp>2007-05-01T17:07:21</time_stamp>
    <author>Thomas Liebech and Nick Nisbet</author>
    <organization>IAI Model Support Group</organization>
    <preprocessor_version>2007 sp3</preprocessor_version>
    <originating_system>Altova XMLSPY</originating_system>
    <authorization>IAI Model Support Group</authorization>
    <documentation>IAI ifcXML implementation guide</documentation>
</iso_10303_28_header>

```

Code 14: De header van een ifcXML bestand

Het data-gedeelte wordt door het “uos”-element weergegeven. Hoewel er meerdere “uos”-elementen in een document mogelijk zijn, stelt het IAI voor om maar één enkel element per document te gebruiken.

```
<uos id="uos_1" description="" configuration="i-ifc2x2" edo=""  
xmlns="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL"  
xmlns:ex="urn:iso.org:standard:10303:part(28):version(2):xmlschema:common"  
xsi:schemaLocation="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL  
http://www.iai-tech.org/ifcXML/IFC2x3/FINAL/IFC2X3.xsd">
```

Code 15: Het data-gedeelte van een ifcXML bestand

IfcXML is momenteel nog niet zo populair als SPF, maar meer en meer toepassingen implementeren ifcXML als alternatief, zoals bijvoorbeeld Graphisoft Archicad. De reden hiervoor is dat met het aanbieden van een XML-weergave van IFC een breder gamma van toepassingen wordt aangesproken en XML de basis is voor de meeste Web Services¹⁵. XML wordt ook ondersteund door de meeste webbrowsers vooral met XSLT style sheets¹⁶, waardoor de informatie onmiddellijk toegankelijk gemaakt wordt.

Hoewel een ifcXML ongeveer 2 à 10 keer groter is dan een SPF, biedt dit bestandsformaat in de toekomst meer slaagkans. Omdat Revit geen ondersteuning biedt voor uitvoer naar ifcXML, wordt dit bestandsformaat in deze thesis niet gebruikt.

¹⁵ Een Web Service is een interface van een applicatie die toegankelijk is via standaard webprotocollen.

¹⁶ XSLT of eXtensible Stylesheet Language Transformations is een standaard voor het omzetten van informatie in een XML-document naar een ander formaat, of naar een XML-document met een andere structuur.

4.4. De structuur van IFC2x

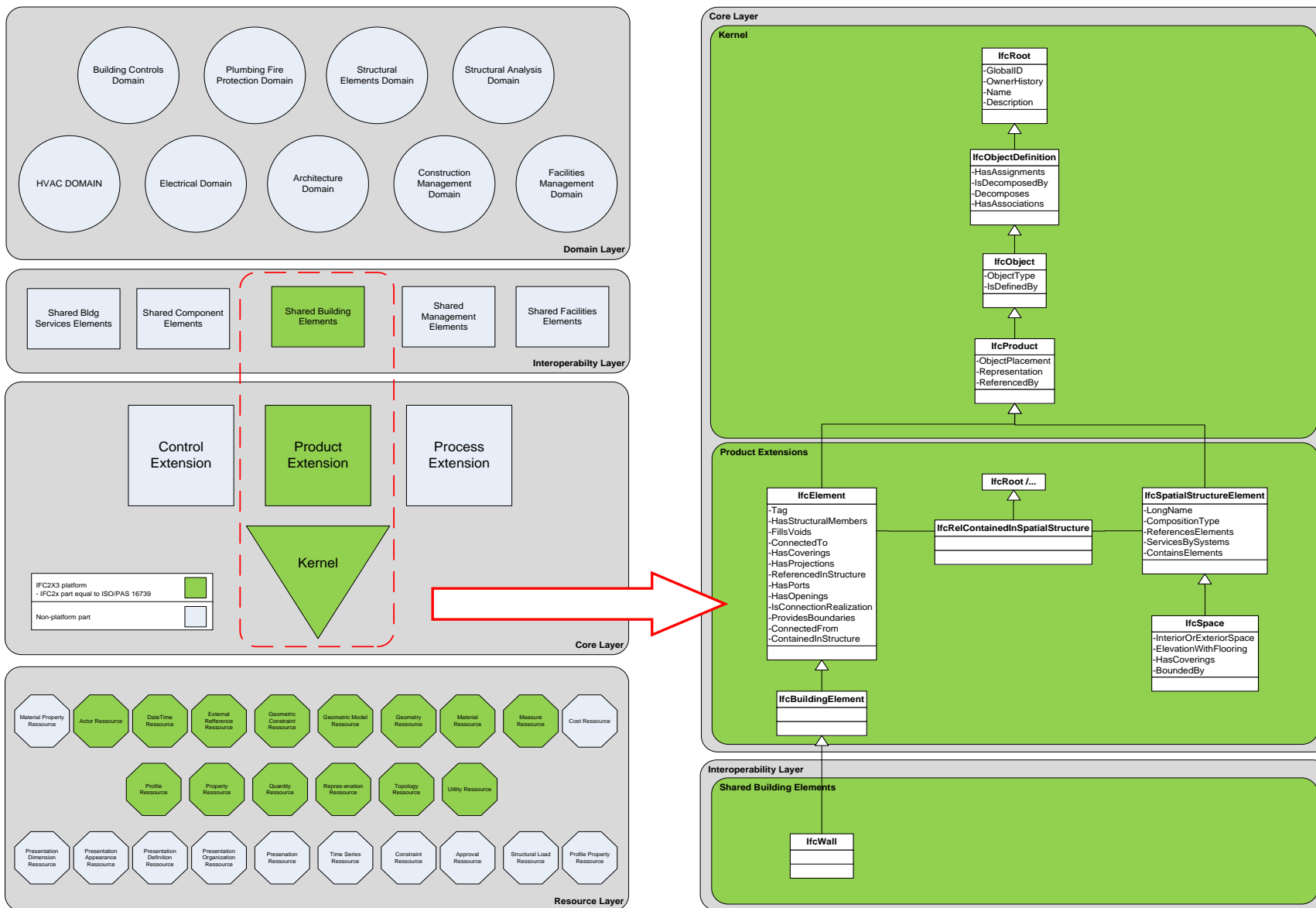
4.4.1 De algemene architectuur

IFC is in voortdurende ontwikkeling en groei. De huidige grote versie van IFC, die in oktober 2000 werd uitgebracht, heet IFC2x. Het belangrijkste doel van deze nieuwe versie is te zorgen voor terugwaatse compatibiliteit tussen alle toekomstige edities van IFC2x. Momenteel is IFC2x toe aan zijn derde editie genaamd IFC2x3. Sinds de release van IFC2x is het aantal entiteiten gegroeid van 370 tot 653 in de huidige uitgave. Om IFC flexibel genoeg te laten omgaan met komende wijzigingen is er voor een meerlaagse, modulaire structuur gekozen. Men onderscheidt vier lagen:

- **Resource Layer:** bevat entiteiten die basiseigenschappen vertegenwoordigen welke niet specifiek zijn voor gebouwen. Bv . geometrie, materiaal, hoeveelheid, afmetingen, datum en tijd, kosten, ...
- **Core Layer:** bevat entiteiten die abstracte concepten vertegenwoordigen. Bv. actor, proces, product, ruimte, gebouw, bouwelement, werkschema, ...
- **Interoperability Layer:** omvat entiteitscategorieën die vaak gebruikt worden over meerdere gebouwen en voorzieningen Bv. Balk, kolom, muur, deur, bewoner, meubilairtype.
- **Domain Layer:** is het hoogste niveau van het IFC-model. Het bevat entiteitdefinities van begrippen die specifiek zijn voor individuele domeinen zoals architectuur, stabiliteit, faciliteitenbeheer.

Elke laag is nog eens onderverdeeld in verschillende categorieën. Elke categorie heeft een definitie voor de verschillende objecten. Bijvoorbeeld een muur (IfcWall) behoort tot Shared Building Elements uit de Interoperability Layer. Een belangrijk kenmerk is dat een object in een laag alleen gerelateerd kan zijn aan een ander object uit dezelfde laag of een lager niveau.

Sommige categorieën in Figuur 17 zijn groen gekleurd, deze categorieën zijn ISO-gecertificeerd. Hierdoor moeten ze voldoen aan bepaalde kwaliteitscontroles. Het rechterdeel van Figuur 17 geeft een voorbeeld van 2 objecten, een muur (IfcWall) en een ruimte (IfcSpace). De IfcWall spreekt voor zich, het is een muur in een gebouw met bijhorende eigenschappen. IfcSpace definieert een ruimte begrensd door bouwelementen. Het gebruik van IfcSpaces maakt het eenvoudig voor allerhande simulaties zoals voor HVAC of brandveiligheid. Het datamodel is gebaseerd op strenge overerving. Dit wordt geïllustreerd in de figuur, waar IfcWall een subtype is van IfcBuildingElement. IfcBuildingElement is dan weer een subtype van IfcElement. Aan de andere kant is IfcSpace een subtype van IfcSpatialStructureElement.



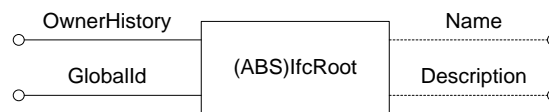
Figuur 17: De algemene architectuur van IFC en een UML-model voorbeeld van IfcWall en IfcSpace. (1)

4.4.2 De Klassen

a) Root klasse

IfcRoot is de meest abstracte en root-klasse voor alle IFC entiteiten die zich in de kernel¹⁷ en daaropvolgende layers bevinden. Het is daarom het supertype van alle entiteiten, behalve die uit de Resource Layer. IfcRoot bepaalt de basisvoorzieningen voor het model:

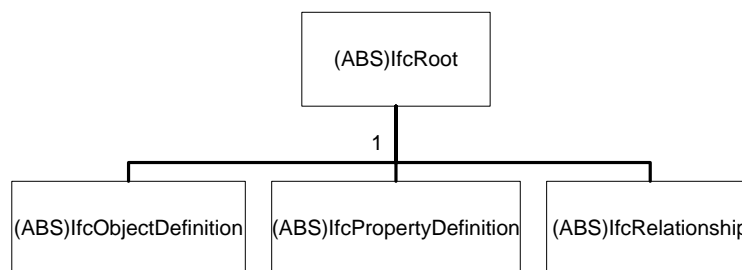
- Object Identiteit
- Naamgeving en beschrijving
- Eigendom en veranderingen



Figuur 18: De IfcRoot klasse

b) Fundamentele klassen

Er zijn 3 fundamentele klassen in het IFC-model, dewelke allemaal zijn afgeleid van IfcRoot. IfcObjectDefinition is de veralgemening van een ding of object. Deze objecten zijn onafhankelijke stukjes informatie die op hun beurt andere informatie kunnen bevatten. IfcRelationship is de generalisatie van alle relaties tussen objecten. Er zijn twee verschillende types van relaties, 1-op1 relaties en 1-op-veel relaties. IfcPropertyDefinition is een verzameling van alle kenmerken, die aan objecten kunnen worden toegekend.



Figuur 19: Fundamentele klassen

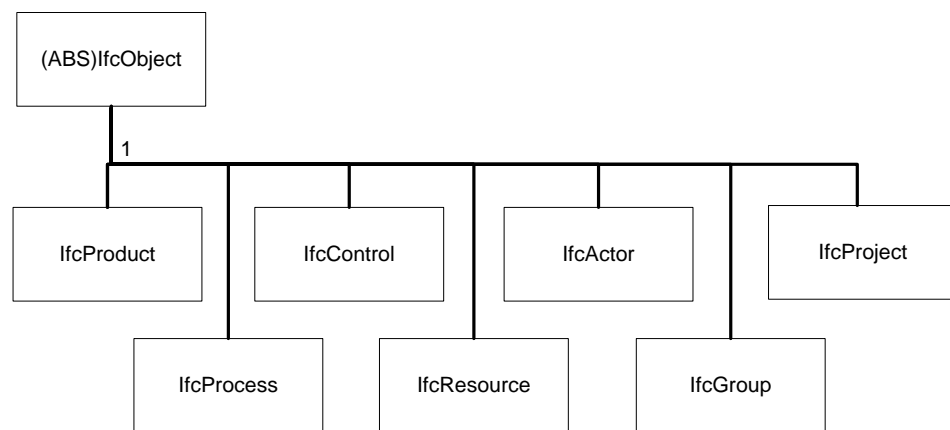
c) Objecten

IfcObject is een subtype van IfcObjectDefinition en bevat zeven fundamentele types:

- IfcProduct: elk object met een geometrische of ruimtelijke context

¹⁷ Een kernel is het meest centrale en onzichtbare onderdeel van een systeem. Het verzorgt alle basisvoorzieningen voor de andere delen uit het systeem.

- IfcProcess: een actie die plaatsvindt in de bouw met betrekking op het ontwerp, de kostprijs of de bouw.
- IfcControl: is de abstracte verzameling van alle concepten die producten of processen controleren
- IfcResource: concept dat het gebruik beschrijft van een object binnen een proces
- IfcActor: definieert alle personen betrokken bij een project tijdens de volledige levenscyclus van een bouwwerk
- IfcProject: basis van een nieuw model
- IfcGroup: een groep is een verzameling van objecten

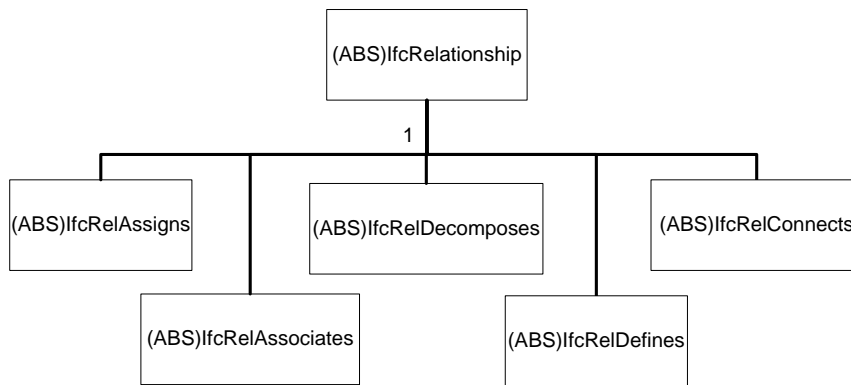


Figuur 20: De klasse IfcObject en zijn subtypes

d) Relaties

Er zijn 5 relatietypes in het IFC model.

- Assignment: verzameling tussen een object en diensten of tussen andere objecten
- Association: legt het verband tussen objecten en externe bronnen (classificatie, bibliotheek of een document)
- Decomposition: definieert een deel of het geheel van een hiërarchie
- Definition: past een definitie van een type of een definitie van een eigenschap toe op een object
- Connection: definieert de connectiviteit tussen objecten



Figuur 21: De klasse IfcRelationship en zijn subtypes

4.5. Ifc Roadmap

Nr	Ifc Release Development	2004				2005				2006				2007				2008				2009				2010			
		K4	K1	K2	K3	K4	K1	K2	K3	K4	K1	K2	K3	K4	K1	K2	K3	K4	K1	K2	K3	K4	K1	K2	K3	K4	K1	K2	K3
1	IFC2x3	[Blue bar]																											
2	IFC2x3 TC1					[Blue bar]																							
3	IFC2x4									[Pink bar]				[Blue bar]															
4	IFC2x4 alpha									[Pink bar]																			
5	IFC2x4 beta1													[Pink bar]															
6	IFC2x4 beta 2																	[Blue bar]											
7	IFC2x4 RC1																					[Blue bar]							

Figuur 22: De roadmap van IFC

De huidige versie van IFC2x, IFC2x3 is in februari 2006 uitgekomen en is momenteel de meest toegepaste versie. In juni 2007 werd er een aanpassing aan IFC2x3 uitgebracht, Technical Corrigendum 1.

De volgende release van IFC2x wordt IFC2x4. De alpha release werd in november 2008 gepubliceerd. Momenteel is de eerste betaversie van IFC2x4 uitgebracht, het werd aan het publiek gepresenteerd op de buildingSMART top in Parijs, midden mei 2009. Een tweede betaversie wordt eind september 2009 op de buildingSMART bijeenkomst in Berlijn verwacht. De Release Candidate zal tegen eind 2009 beschikbaar zijn en zal waarschijnlijk als definitief worden aangegeven wanneer de eerste certificering begonnen is.

4.6. Certificering

IAI biedt de mogelijkheid om toepassingen die IFC ondersteunen te certificeren. Certificaten moeten volgens bepaalde regels worden uitgevoerd en zullen na afloop van de certificering publiek worden gemaakt. Een certificering gebeurt in twee stappen:

- Stap 1: de applicatie wordt met een reeks van testbestanden tijdens een openbare workshop getest.

- Stap 2: nadat eindgebruikers de applicatie voor ten minste 6 maanden gebruikt hebben, zal de applicatie getest worden met behulp van gegevens van reële projecten.

Na elke stap wordt er een certificaat afgeleverd. Sinds begin februari bedraagt de huidige tarifiering voor de IFC2x3-certificering:

- USD 2000 : voor toegang tot de volledige set van testbestanden. Een klein deel van de testbestanden zijn beschikbaar op de CD-rom of op de website: <http://www.iai.hm.edu/>
- USD 1500 : voor deelname aan de eerste stap van de certificering
- USD 1500 : voor deelname aan de tweede stap van de certificering

Het bedrag voor een volledige certificering komt dan neer op USD 5000. Wanneer applicaties met succes een certificering doorstaan wordt er een logo verleend aan de softwareproducent. De logo's mogen gebruikt worden op hun website, in publicaties en op de verpakkingen hun van software. In Figuur 23 staat links het logo voor de "Eerste Stap"-certificatie, rechts het logo voor de "Tweede Stap"-certificatie.



Figuur 23: Certificering Logo's (8)

Volgende applicaties hebben officieel de volledige certificering doorlopen:

Tabel 1: Bedrijven bekroond met het logo van de 2^{de} stap certificering

Product	Bedrijf	Beperking	Datum
ACTIVE3d	ARCHIMEN Group	Alleen invoer	13-03-07
ALLPLAN 2006.2	Nemetschek		13-03-07
ArchiCAD 11	Graphisoft		13-03-07
AutoCAD Architecture 2008 SP1	Autodesk		13-03-07
Bentley Architecture 8.9.3	Bentley Systems		13-03-07
DDS-CAD 6.4	DDS		13-03-07
Facility Online	Vizelia	Alleen invoer	22-05-07
MagiCAD	Progman	Alleen uitvoer	22-05-07
ESA-PT	SCIA		25-02-08

Revit Building 2008 SP1	Autodesk		13-03-07
Solibri Model Checker	Solibri	Alleen invoer	13-03-07
TEKLA Structures	TEKLA Corporation		13-03-07

4.7. Nieuwigheden in IFC2x4

De belangrijkste vernieuwing in IFC2x4 moet de toevoeging van de volledige ondersteuning voor de IFD Library (zie 3.4.2) worden.

Onderstaand voorbeeld legt uit hoe een IFD Library gekoppeld kan worden aan een object. `IfcRelAssociatesLibrary` legt het verband tussen `IfcLibraryReference` en een venster (`IfcWindow`). `IfcLibraryReference` is de verwijzing naar een bibliotheek door middel van een locatie (zoals een URL¹⁸). Extra informatie over de bibliotheek wordt aan `IfcLibraryReference` meegegeven door de `IfcLibraryInformation`, zoals de naam, versie, datum en uitgever van de bibliotheek.

```
#75232=IFCLIBRARYINFORMATION('IFD','2.0',#75234,#75238,$);
#75234=IFCORGANIZATION('SINTEF','SINTEF Byggforsk','SINTEF building and
infrastructure',,$,$);
#75238=IFCCALENDARDATE(2,4,2008);
#75239=IFCLIBRARYREFERENCE('http://buildingsmart.byggforsk.no/ifdviewer/Main.html
?ifdguid=3vHdeKoT0Hsm00051Mm008','3vHdeKoT0Hsm00051Mm008','aluminiumsvindu','nb-
NO',#75232);
#75241=IFCRELASSOCIATESLIBRARY('1KOS200MKHtG00051Mm008',#75242,$,$,(#3284),#75239
);
#75242=IFCOWNERHISTORY($,$,.READWRITE.,.NOCHANGE.,1207310661,$,$,1207310661);
#3284=IFCWINDOW('21t9JAm_r44uLZ7bhWHGny',#21,'001',,$,$,#1467,#3281,$,1200.,1600.)
;
```

Code 16: Voorbeeld van het gebruik van een IFD Library voor een venster

Bouwelementen beschikken nu ook over een “standaard geval”. Dit was al het geval voor muren (`IfcWall/IfcWallStandardCase`), maar dit is nu ook van toepassing bij balken, kolommen, deuren, vensters en vloeren. De schoorsteen is een nieuw bouwelement.

Materialen hebben nu ook een beschrijving en categorie, zodat ze ingedeeld kunnen worden volgens die categorie. De eigenschappen hebben een opfrisbeurt gekregen, ze zijn vereenvoudigd en moeten gemakkelijker aan een materiaal verbonden kunnen worden.

Verder werden er een heleboel verbeteringen aangebracht, zoals aan het elektriciteitsdomein en zijn een massa aan bugs opgelost.

¹⁸ Een Uniform Resource Locator, afgekort URL, is een label dat verwijst naar een informatiebron, bijvoorbeeld een webpagina of een ander bestand.

5. Analyse van het informatiemodel

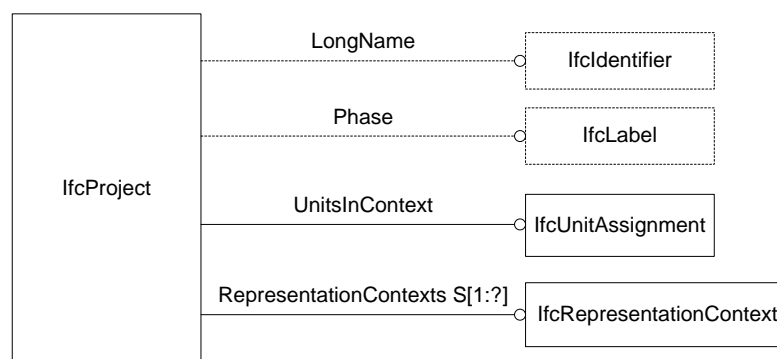
In dit deel wordt de opbouw van een model, de mogelijke geometrische voorstellingen van basis bouwelementen en de weergave van materialen besproken. Dit hoofdstuk is gebaseerd op de documentatie van IFC2x3 TC1 (9).

5.1. Opbouw van het Model

5.1.1 Ifcproject

Het project verbindt een aantal ontwerp-, bouw- of onderhoudsactiviteiten tot een product. Het project voorziet het model van primaire informatie. Die informatie bevat:

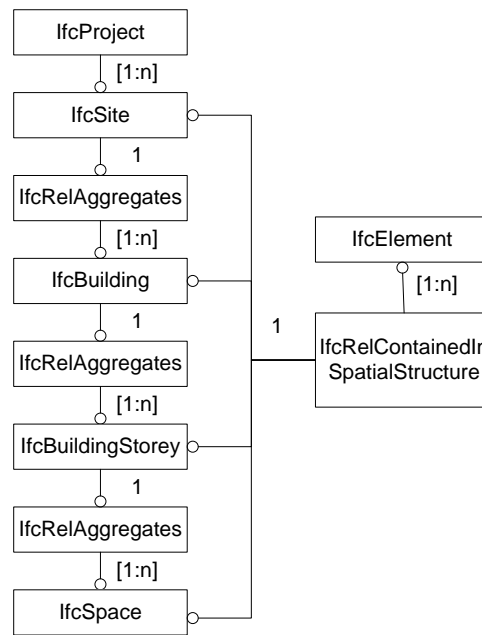
- De gebruikte eenheden (UnitsInContext)
- Het globale assenkruis
- De precisie waarmee getekend wordt
- Eventueel de vermelding van het ware noorden ten opzichte van het globale assenkruis



Figuur 24: De klasse IfcProject

Aan de basis van een model staat het IfcProject. IfcProject is als het ware de wortel van de boomstructuur. Er kan dus maar één IfcProject per bestand mogelijk zijn. De ruimtelijke structurelementen zijn aan het IfcProject verbonden met behulp van een IfcRelAggregates relatie. Het project verwijst naar deze elementen door haar inverse relatie:

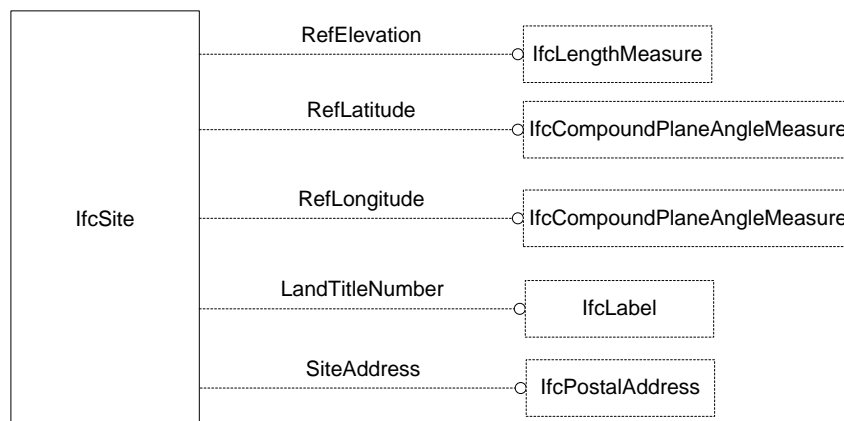
- IfcProject.Decomposes: is gelijk aan nul, IfcProject is de basis van het model en is dus geen ontleding van een ander object
- IfcProject.IsDecomposedBy: verwijst naar een IfcSite of IfcBuilding, deze vormen de basis van de ruimtelijke structuur



Figuur 25: Ruimtelijke Structuur

5.1.2 IfcSite

IfcSite definieert een afgebakend gebied waarop het project wordt gebouwd. De locatie van de site, gedefinieerd door IfcLocalPlacement, is altijd relatief ten opzichte van het ruimtelijk element waarin deze site is opgenomen, of absoluut wanneer het vorige element het IfcProject is.



Figuur 26: De klasse IfcSite

Een project kan bestaan uit meerdere met elkaar verbonden of aparte sites. Daarom voorziet het complex in een verzameling van sites. Een site kan ook opgedeeld worden in kleinere delen. In totaal onderscheidt men drie verschillende sites:

- COMPLEX: het ganse site complex
- PARTIAL: een site-onderdeel
- ELEMENT: een site

Net zoals bij `IfcProject` verwijst `IfcSite` via inverse relaties naar andere ruimtelijke structuren:

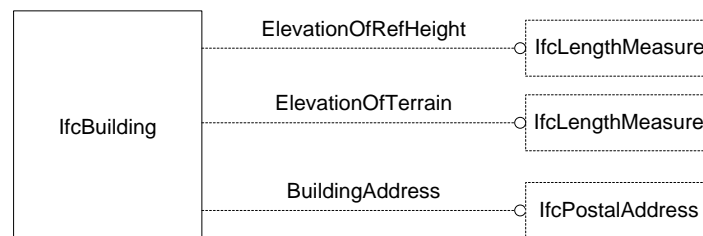
- `IfcSite.Decomposes`: verwijst naar het `IfcProject` of naar de `IfcSite` van een hoger niveau, dat wil zeggen naar een site van niveau `COMPLEX` wanneer het niveau `ELEMENT` is of naar een site van niveau `ELEMENT` wanneer het niveau `PARTIAL` is
- `IfcSite.IsDecomposedBy`: verwijst naar een `IfcSite`, `IfcBuilding` of `IfcSpace`

Als er bouwelementen of andere elementen direct verbonden zijn met de site, dan worden ze aan de site gekoppeld via de relatie `IfcRelContainedInSpatialStructure`.

5.1.3 IfcBuilding

Een gebouw wordt meestal gebouwd op een site. Een gebouw kan bestaan uit meerdere aaneengesloten of losse gebouwen. Daarom bevat een complex een verzameling van meerdere gebouwen in een site. Een gebouw kan ook ontleed worden in (verticale) delen, waarbij elk deel een bouwdeel wordt. In totaal onderscheidt men drie verschillende gebouwen:

- `COMPLEX`: het ganse complex
- `PARTIAL`: een bouwdeel
- `ELEMENT`: een gebouw



Figuur 27: De klasse `IfcBuilding`

`IfcBuilding` verwijst naar andere ruimtelijke elementen via de relatie `IfcRelAggregates`:

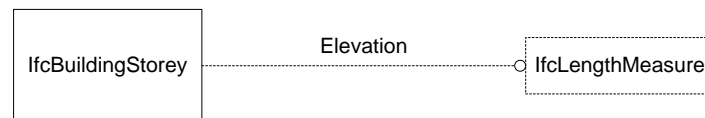
- `IfcBuilding.Decomposes`: verwijst naar `IfcSite` of `IfcBuilding` van een hoger niveau, dat wil zeggen naar een gebouw van niveau `COMPLEX` wanneer het niveau `ELEMENT` is of naar een gebouw van niveau `ELEMENT` wanneer het niveau `PARTIAL` is
- `IfcBuilding.IsDecomposedBy`: verwijst naar een `IfcBuilding` of `IfcBuildingStorey`

Als er bouwelementen of andere elementen direct verbonden zijn met het gebouw, dan worden ze aan het gebouw gekoppeld via de relatie `IfcRelContainedInSpatialStructure`.

5.1.4 IfcBuildingStorey

Een verdieping is verbonden aan een gebouw. Een verdieping kan opgedeeld worden in kleinere (horizontale) delen. In totaal onderscheidt men drie verschillende types verdiepingen:

- COMPLEX: het ganze complex
- PARTIAL: een deel van een verdieping
- ELEMENT: een verdieping



Figuur 28: De klasse IfcBuildingStorey

IfcBuildingStorey verwijst naar andere ruimtelijke elementen via IfcRelAggregates:

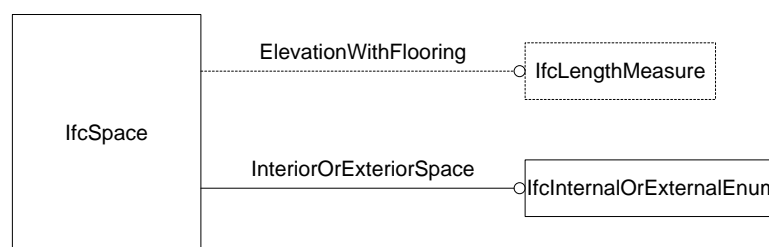
- IfcBuildingStorey.Decomposes: verwijst naar IfcBuilding of IfcBuildingStorey van een hoger niveau, dat wil zeggen naar een verdieping van niveau COMPLEX wanneer het niveau ELEMENT is of naar een verdieping van niveau ELEMENT wanneer het niveau PARTIAL is
- IfcBuildingStorey.IsDecomposedBy: verwijst naar een IfcBuildingStorey of IfcSpace

Als er bouwelementen of andere elementen direct verbonden zijn met de verdieping, dan worden ze aan de verdieping gekoppeld via de relatie IfcRelContainedInSpatialStructure.

5.1.5 IfcSpace

Een ruimte geeft een oppervlakte of volume weer die bepaalde functies binnen een gebouw uitoefent. Een ruimte is aan een verdieping verbonden, of aan een site in geval van buitenruimtes. Een ruimte kan ontleed worden in deelruimtes. In totaal onderscheidt men drie verschillende types ruimtes:

- COMPLEX: een groep van ruimtes
- PARTIAL: een deelruimte
- ELEMENT: een ruimte



Figuur 29: De klasse IfcSpace

IfcSpace verwijst naar andere ruimtelijke elementen via zijn inverse relatie IfcRelAggregates:

- IfcSpace.Decomposes: verwijst naar IfcSite, IfcBuildingStorey of IfcSpace van een hoger niveau, dat wil zeggen naar een ruimte van niveau COMPLEX wanneer het niveau ELEMENT is of naar een ruimte van niveau ELEMENT wanneer het niveau PARTIAL is
- IfcSpace.IsDecomposedBy: verwijst naar een IfcSpace

Als er bouwelementen of andere elementen direct verbonden zijn met de ruimte, dan worden ze aan de ruimte gekoppeld via de relatie IfcRelContainedInSpatialStructure.

5.2. Geometrische voorstelling

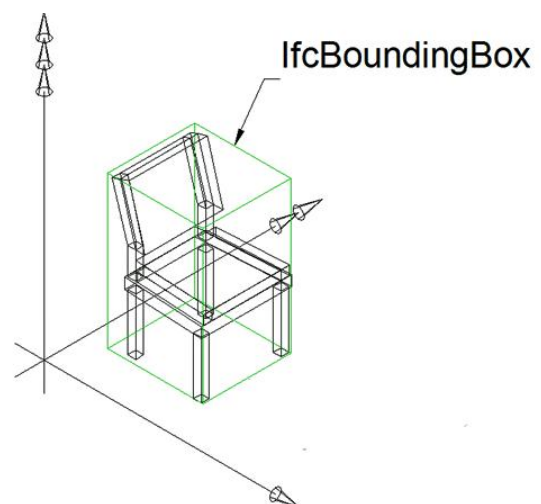
5.2.1 Algemeen

De geometrische voorstelling van alle bouwelementen (IfcBuildingElement) wordt weergegeven door IfcProductDefinitionShape en IfcLocalPlacement. De locatie voor een IfcBuildingElement is omschreven in haar supertype IfcProduct. Zij wordt gedefinieerd door een IfcLocalPlacement, dat naar het lokale coördinatensysteem van de geometrische voorstelling verwijst. Verdere beperkingen zijn vastgesteld op het niveau van zijn subtypen.

Elke IfcBuildingElement kan één of meerdere geometrische voorstellingen bevatten. Alleen de algemene types 'BoundingBox', 'SurfaceModel', 'Brep' en 'MappedRepresentation' worden hieronder verder uitgelegd.

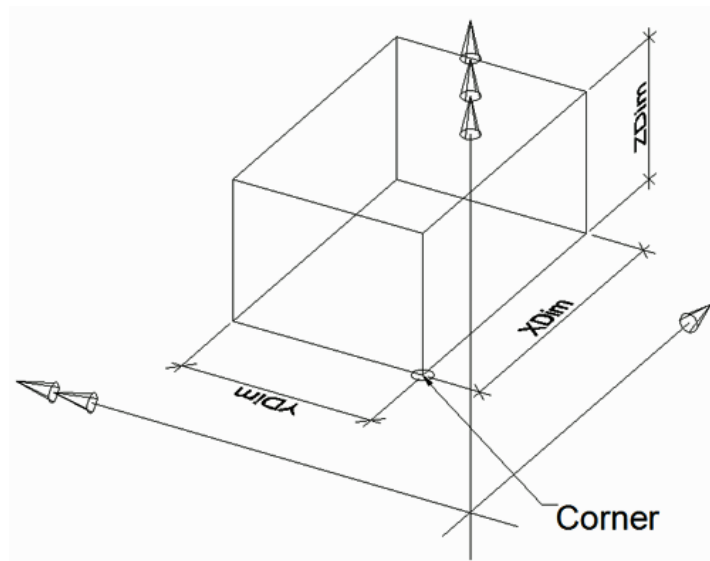
5.2.1.1. BoundingBox

Een bouwelement kan voorgesteld worden als een bounding box. Dit is een doos met de kleinste afmetingen dat het hele element omhuld. Een bounding box is de eenvoudigste geometrische voorstelling die beschikbaar is.



Figuur 30: Bounding Box (9)

De `IfcBoundingBox` wordt bepaald door de linker benedenhoek (`Corner`) en de rechter bovenhoek, die te bepalen is aan de hand van `XDim`, `YDim` en `ZDim`.



Figuur 31: Definitie van een `IfcBoundingBox` (9)

De volgende kenmerken dienen voor de `IfcShapeRepresentation` gebruikt te worden:

- `RepresentationIdentifier`: 'Box'
- `RepresentationType`: 'BoundingBox'

```
#47=IFCBOUNDINGBOX(#46,6803.,2965.,2750.);
#48=IFCSHAPEREPRESENTATION(#11,'Box','BoundingBox',(#47));
```

Code 17: Gebruik van een bounding box in IFC

5.2.1.2. *SurfaceModel*

Een bouwelement kan voorgesteld worden als een enkelvoudig of een meervoudig oppervlak. Een `SurfaceModel` kan bestaan uit een `IfcShellBasedSurfaceModel` of uit een `IfcFaceBasedSurfaceModel`.

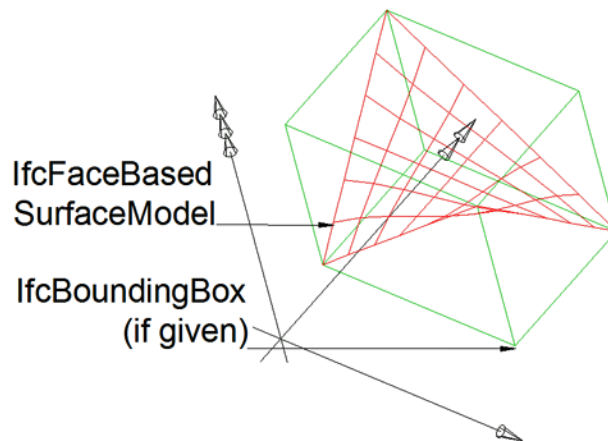
De volgende kenmerken dienen voor de `IfcShapeRepresentation` gebruikt te worden:

- `RepresentationIdentifier`: 'Body'
- `RepresentationType`: 'SurfaceModel'

```
#496=IFCCONNECTEDFACESET((#86,#93,#174,#179,#184,#323,#489,#492,#495));
#497=IFCFACEBASEDSURFACEMODEL((#496));
#498=IFCSHAPEREPRESENTATION(#27,'Body','SurfaceModel',(#497));
```

Code 18: Gebruik van een `SurfaceModel` in IFC

In sommige gevallen kan het nuttig zijn om een geometrie eenvoudiger voor te stellen aan de hand van een bounding box.



Figuur 32: SurfaceModel met Bounding Box (9)

5.2.1.3. Brep

Een bouwelement kan voorgesteld worden als een enkelvoudige of meervoudige “Boundary Representation” elementen. Boundary Representation of BREP is een methode om volumes voor te stellen met behulp van grenzen. Een volume is een verzameling van aaneengesloten vlakken. Zulke vlakken kunnen op verschillende manieren worden opgebouwd. In IFC gebeurt dit enkel aan de hand van zogenaamde “faces”. Een “face” is een begrensd vlak. Deze voorstelling dient voor elementen met een complexe geometrische vorm. De volgende kenmerken dienen voor de `IfcShapeRepresentation` gebruikt worden:

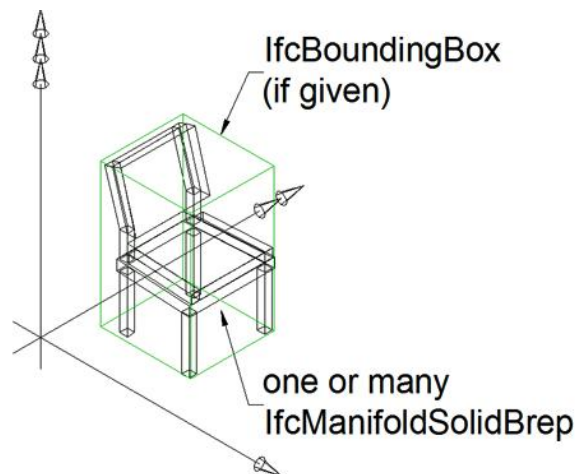
- RepresentationIdentifier: 'Body'
- RepresentationType: 'Brep'

```
#123=IFCCLOSEDSHELL((#103,#110,#113,#116,#119,#122));
#124=IFCFACETEDBREP(#123);
#125=IFCSHAPEREPRESENTATION(#18,'Body','Brep',(#124));
```

Code 19: Gebruik van Brep in IFC

Brep voorstelling wordt gegeven door een `IfcShapeRepresentation` met één of meer items van het type `IfcManifoldSolidBrep`.

In sommige gevallen kan het nuttig zijn om een eenvoudige voorstelling ook als een bounding box voor te stellen van dezelfde complexe vorm. In Figuur 33 wordt een stoel vervangen door een omhullende doos.



Figuur 33: Brep met Bounding Box (9)

5.2.1.4. Mappedrepresentation

Elk bouwelement kan voorgesteld worden als een Mappedrepresentation. De volgende kenmerken dienen voor de IfcShapeRepresentation gebruikt te worden:

- RepresentationIdentifier: 'Body'
- RepresentationType: 'MappedRepresentation'

5.2.2 Muren

Momenteel wordt het gebruik van 'SweptSolid'-, 'Clipping'- en 'Brep'-voorstellingen voor muren ondersteund. Naast deze algemene voorstellingen zijn de types 'SurfaceModel' en 'Bounding Box' ook toegestaan. SweptSolid en Clipping zal hier verder uitgelegd worden, Brep is hierboven al beproven.

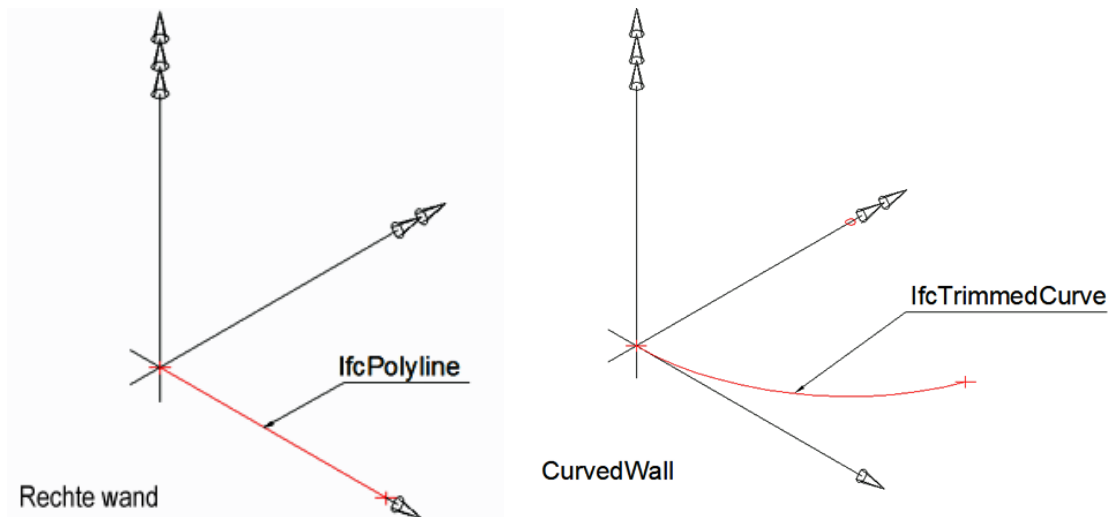
De geometrische voorstelling kan opgedeeld worden in twee delen, de as van de muur en het lichaam van de muur.

Eerste voorstelling: Curve2D voorstelling van de as van de muur

De as van de muur wordt door een tweedimensionale open curve voorgesteld. De as dient om enkele eigenschappen aan een muur te koppelen, bijvoorbeeld de opbouw van de materiaallagen. De volgende kenmerken dienen voor de IfcShapeRepresentation gebruikt te worden:

- RepresentationIdentifier: 'Axis'
- RepresentationType: 'Curve2D'

Bij een rechte muur wordt de as voorgesteld door een IfcPolyline of een IfcTrimmedCurve waarvan de BasisCurve gelijk is aan IfcLine. In het geval van een gebogen muur wordt de as voorgesteld door een IfcTrimmedCurve waarvan de BasisCurve gelijk is aan een IfcCircle.



Figuur 34: De voorstelling van de as van een rechte en een gebogen muur (9)

```
#4=IFCCARTESIANPOINT((0.,0.));
#46=IFCCARTESIANPOINT((12300.,0.));
#47=IFCPOLYLINE((#4,#46));
#48=IFCSHAPEREPRESENTATION(#27,'Axis','Curve2D',(#47));
```

Code 20: Curve2D voorstelling van een rechte muur

Tweede voorstelling: SweptSolid of Clipping voorstelling van het muurlichaam

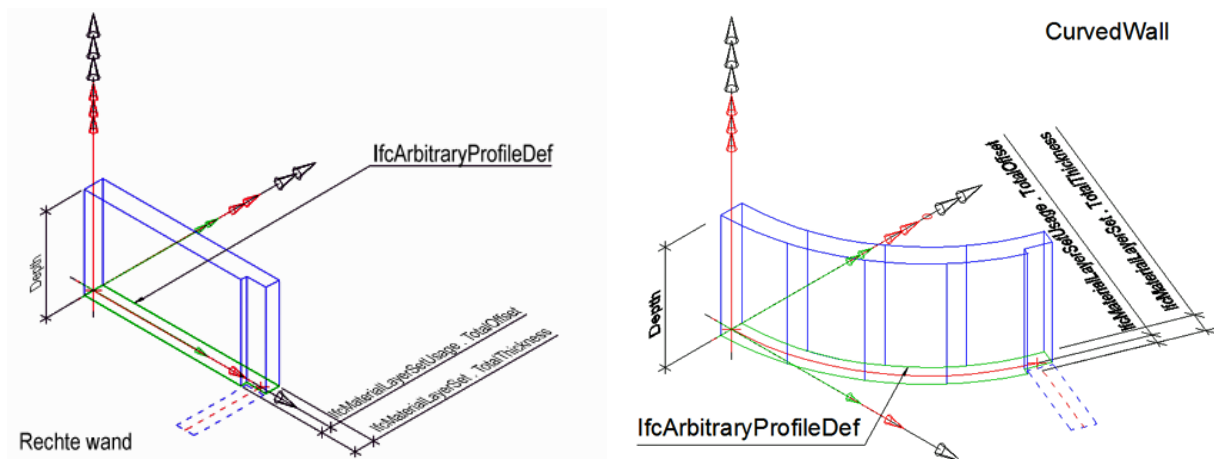
Het lichaam van de muur is gedefinieerd als SweptSolid voor wanden zonder snijvlakken of Clipping voor wanden met snijvlakken (bv. onder een hellend dak).

IfcShapeRepresentation heeft de volgende waarden:

- RepresentationIdentifier: 'Body'
- RepresentationType: 'SweptSolid' of 'Clipping'

Wanneer de standaard geometrische voorstelling van een muur gedefinieerd is als SweptSolid, zijn de volgende beperkingen van toepassing:

- Solid: IfcExtrudedAreaSolid
- Profiel: IfcArbitraryClosedProfile of IfcRectangleProfileDef
- Extrusie: Het profiel wordt in de richting van de z-as geëxtrudeerd, de z-as staat loodrecht op het profiel.



Figuur 35: SweptSolid van een rechte en gebogen wand

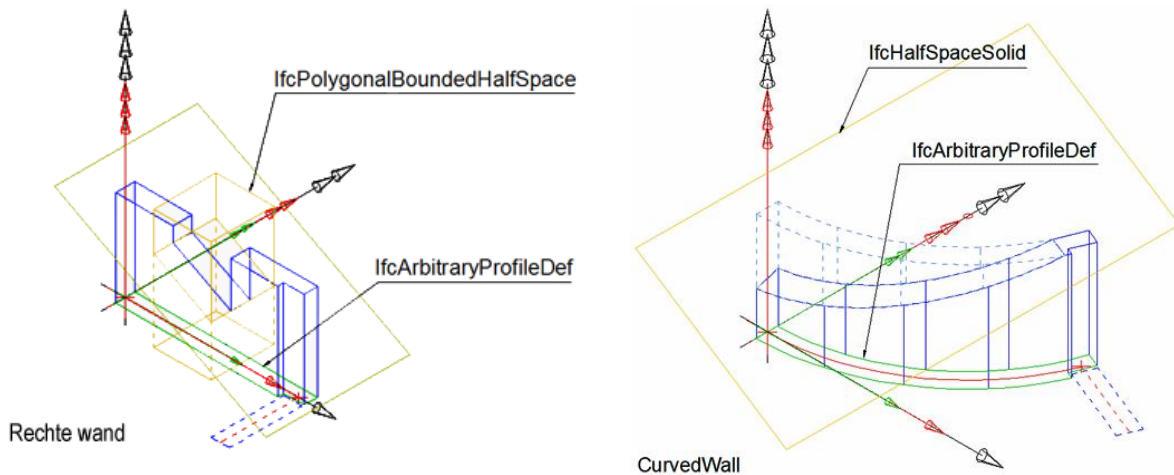
```

#51=IFCCARTESIANPOINT((0.,0.24));
#52=IFCCARTESIANPOINT((0.,0.));
#53=IFCCARTESIANPOINT((8.,0.));
#54=IFCCARTESIANPOINT((8.,0.24));
#55=IFCCARTESIANPOINT((0.,0.24));
#56=IFCPOLYLINE((#51,#52,#53,#54,#55));
#57=IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,$,#56);
#58=IFCCARTESIANPOINT((0.,0.,0.));
#59=IFCDIRECTION((0.,0.,1.));
#60=IFCDIRECTION((1.,0.,0.));
#61=IFCAXIS2PLACEMENT3D(#58,#59,#60);
#62=IFCDIRECTION((0.,0.,1.));
#63=IFCEXTRUDEDAREASOLID(#57,#61,#62,3.);
#65=IFCSHAPEREPRESENTATION(#11,'Body','SweptSolid',(#63));
    
```

Code 21: SweptSolid van een rechte wand

De meer complexe en complete geometrische voorstelling van een wand is Clipping. Hierbij zijn volgende beperkingen van toepassing:

- Solid: *IfcExtrudedAreaSolid*
- Profiel: *IfcArbitraryClosedProfile* of *IfcRectangleProfileDef*
- Extrusie: Het profiel wordt in de richting van de z-as geëxtrudeerd, de z-as staat loodrecht op het profiel.
- Boolean resultaat (*IfcBooleanClippingResult*): Boole verschil tussen de extrusie en één of meerdere snijvlakken (*IfcHalfSpaceSolids*)



Figuur 36: Clipping van een rechte en een gebogen wand (9)

```
#49=IFCCARTESIANPOINT((6150.,0.));
#50=IFCAXIS2PLACEMENT2D(#49,#12);
#51=IFCRECTANGLEPROFILEDEF(.AREA.,$, #50,12300.00000000002,200.);
#52=IFCAXIS2PLACEMENT3D(#3,$,$);
#53=IFCEXTRUDEDAREASOLID(#51,#52,#9,4000.);
#59=IFCCARTESIANPOINT((12300.,-0.));
#60=IFCCARTESIANPOINT((12300.,200.));
#61=IFCCARTESIANPOINT((12100.,200.));
#62=IFCCARTESIANPOINT((8400.,200.));
#63=IFCCARTESIANPOINT((8200.,200.));
#64=IFCCARTESIANPOINT((200.,200.));
#65=IFCCARTESIANPOINT((0.,200.));
#66=IFCPOLYLINE((#4,#59,#60,#61,#62,#63,#64,#65,#4));
#67=IFCCARTESIANPOINT((0.,-100.,4000.));
#68=IFCAXIS2PLACEMENT3D(#67,#9,#6);
#69=IFCPLANE(#68);
#70=IFCCARTESIANPOINT((0.,-100.,4000.));
#71=IFCAXIS2PLACEMENT3D(#70,$,$);
#72=IFCPOLYGONALBOUNDEDHALFSPACE(#69,.F.,#71,#66);
#73=IFCBOOLEANCLIPPINGRESULT(.DIFFERENCE.,#53,#72);
#74=IFCSHAPEREPRESENTATION(#27,'Body','Clipping',(#73));
```

Code 22: Clipping van een rechte wand

5.2.3 Vloeren

Momenteel wordt het gebruik van 'SweptSolid'-, 'Clipping'-, 'Brep'- en 'MappedRepresentation'-voorstellingen voor vloeren ondersteund. Daarnaast is het gebruik van 'Bounding Box' ook toegestaan. Enkel SweptSolid en Clipping worden verder besproken.

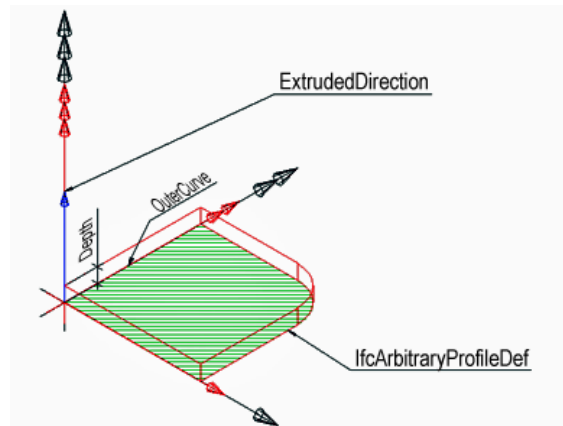
SweptSolid

IfcShapeRepresentation heeft de volgende waarden:

- RepresentationIdentifier: 'Body'
- RepresentationType: 'SweptSolid'

De standaard geometrische voorstelling van een vloer is gedefinieerd als SweptSolid en volgende beperkingen zijn hier dan van toepassing:

- Solid: IfcExtrudedAreaSolid
- Profiel: IfcArbitraryClosedProfile, IfcCircleProfileDef, IfcEllipseProfileDef of IfcRectangleProfileDef
- Extrusie: Het profiel kan al dan niet loodrecht geëxtrudeerd worden



Figuur 37: SweptSolid van een vloer (9)

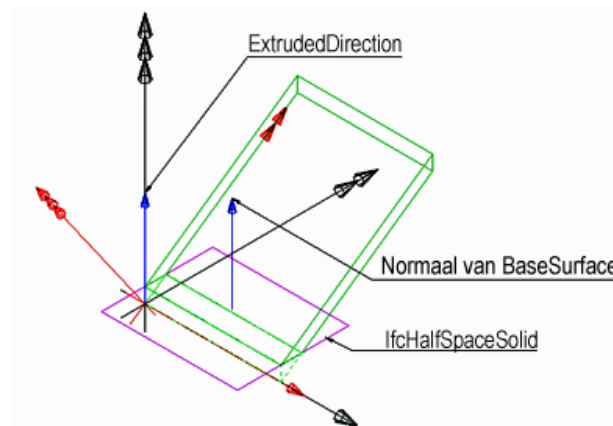
Clipping

IfcShapeRepresentation heeft de volgende waarden:

- RepresentationIdentifier: 'Body'
- RepresentationType: 'Clipping'

De standaard geometrische voorstelling van een vloer is dan gedefinieerd als Clipping. Hierbij zijn volgende beperkingen van toepassing:

- Solid: IfcExtrudedAreaSolid
- Profiel: IfcArbitraryClosedProfile, IfcCircleProfileDef, IfcEllipseProfileDef of IfcRectangleProfileDef
- Extrusie: Het profiel wordt al dan niet loodrecht
- Boolean resultaat (IfcBooleanClippingResult): Boole verschil tussen de extrusie en één of meerdere snijvlakken (IfcHalfSpaceSolids)



Figuur 38: Clipping van een vloer (9)

5.2.4 Balken

Het gebruik van 'SweptSolid'-, 'Clipping'- en 'MappedRepresentation'-voorstellingen is voor balken toegestaan. Daarnaast is het gebruik van 'Brep', 'SurfaceModel' en 'Bounding Box' ook toegelaten.

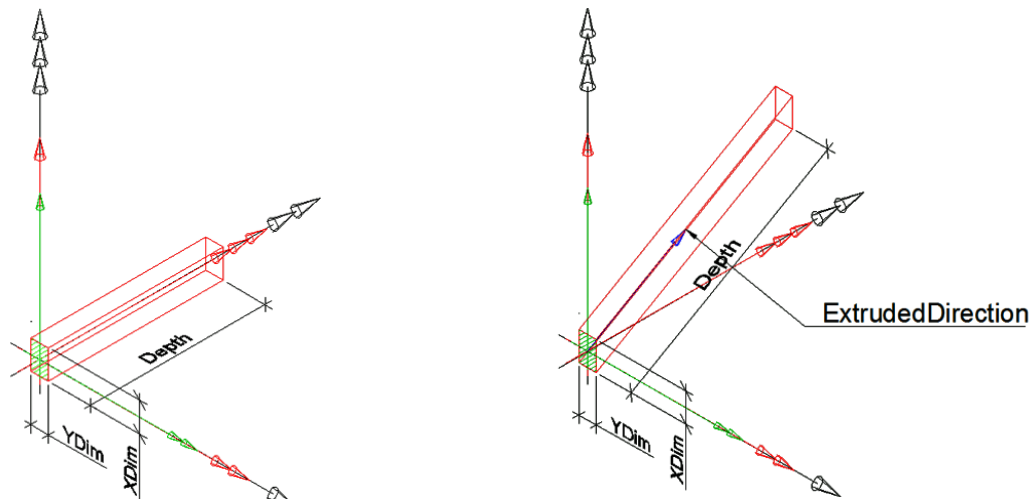
SweptSolid

IfcShapeRepresentation heeft de volgende waarden:

- RepresentationIdentifier: 'Body'
- RepresentationType: 'SweptSolid'

De standaard geometrische voorstelling van een vloer is gedefinieerd als SweptSolid. Hierbij zijn volgende beperkingen van toepassing:

- Solid: IfcExtrudedAreaSolid
- Profiel: IfcArbitraryClosedProfile, IfcCircleProfileDef, IfcEllipseProfileDef of IfcRectangleProfileDef, het gebruik van de basis stalen profielen zijn ook toegestaan, zoals IfcIShapeProfileDef, IfcLShapeProfileDef, IfcUShapeProfileDef, IfcCShapeProfileDef, IfcZShapeProfileDef en IfcTShapeProfileDef
- Extrusie: alle richtingen zijn toegelaten



Figuur 39: SweptSolid van een balk (9)

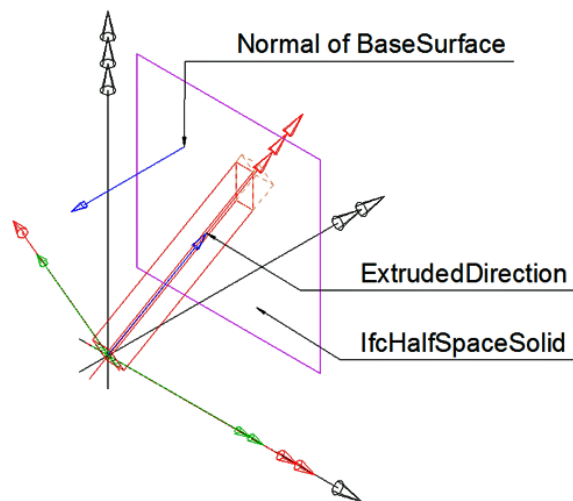
Clipping

IfcShapeRepresentation heeft de volgende waarden:

- RepresentationIdentifier: 'Body'
- RepresentationType: 'Clipping'

De standaard geometrische voorstelling van een balk is gedefinieerd als Clipping. Hierbij zijn volgende beperkingen van toepassing:

- Solid: IfcExtrudedAreaSolid
- Profiel: IfcArbitraryClosedProfile, IfcCircleProfileDef, IfcEllipseProfileDef of IfcRectangleProfileDef, het gebruik van de basis stalen profielen zijn ook toegestaan, zoals IfcIShapeProfileDef, IfcLShapeProfileDef, IfcUShapeProfileDef, IfcCShapeProfileDef, IfcZShapeProfileDef en IfcTShapeProfileDef
- Extrusie: Het profiel kan geëxtrudeerd worden loodrecht of niet loodrecht op het profiel.
- Boolean resultaat (IfcBooleanClippingResult): Boole verschil tussen de extrusie en één of meerdere snijvlakken (IfcHalfSpaceSolids)



Figuur 40: Clipping van een balk (9)

5.2.5 Vensters en Deuren

Er zijn meerdere voorstellingen mogelijk voor de geometrie van vensters en deuren.

- **Profiel:** een `GeometricCurveSet`, bestaande uit een gesloten kromme die de buitenste rand van een venster of deur bepaalt.

```
#30031= IFCSHAPEREPRESENTATION(#111, 'FootPrint', 'GeometricCurveSet',
(#30032));
#30032= IFCGEOMETRICCURVESET((#30087, #30094, #30101, #30108, #30115,
#30122, #30129, #30136, #30143, #30150, #30157));
#30083= IFCCARTESIANPOINT((-4000., 0.));
#30085= IFCCARTESIANPOINT((28000., 0.));
#30087= IFCPOLYLINE((#30083, #30085));
```

Code 23: Profile van een deur

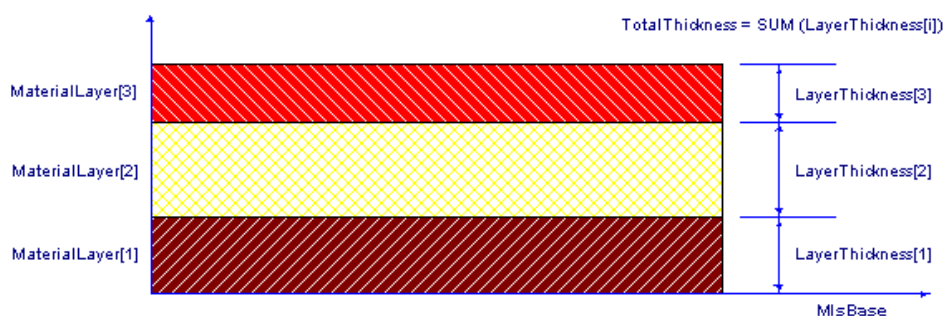
- **Footprint:** een `GeometricCurveSet` of een `Annotation2D` voorstelling van de tweedimensionale vorm van de venster of de deur. De Footprint wordt gebruikt voor de weergave van een venster of deur op een grondplan.
- **Body:** een `SweptSolid`, `Surfacemodel`, of `Brep` van de driedimensionale vorm van de venster of de deur. Body wordt gebruikt voor de weergave van een venster of deur in het model.

5.3. Materialen

Een materiaal wordt aan een bouwelement gekoppeld door de `IfcRelAssociatesMaterial` relatie. Een materiaal kan bestaan uit:

- Één enkel materiaal (IfcMaterial)
- Een lijst van materialen (IfcMaterialList), bijvoorbeeld voor composiet elementen. De klasse zal gewoonlijk worden gebruikt wanneer een element op een meer abstract niveau wordt beschreven. Het mag niet gebruikt worden voor elementen met materiaallagen wanneer deze verschillende lagen duidelijk gedefinieerd kunnen worden. Ook mag IfcMaterialList niet gebruikt worden voor elementen met één enkel materiaal.
- Een verzameling van materiaallagen (IfcMaterialLayerSet) met een richting en een laagdikte. Een IfcMaterialLayerSet definieert impliciet een referentielijn (MlsBase). De eerste laag wordt vanaf deze referentielijn geplaatst en de volgende lagen worden bovenop de vorige laag geplaatst. Er mogen geen gaten of overlappingsen voorkomen. De totale dikte van een laag wordt berekend uit de afzonderlijke laagdiktes.

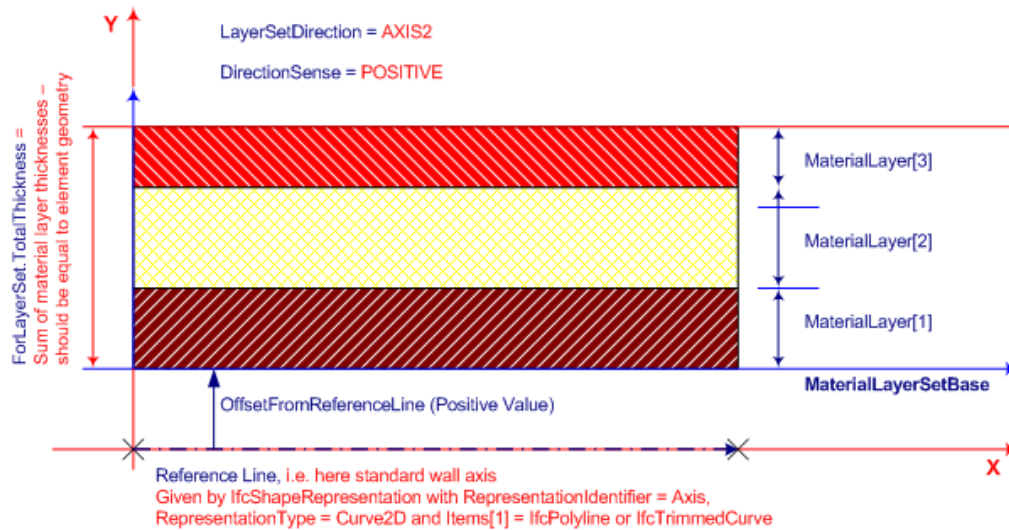
Voorbeeld: een spouwmuur kan gedefinieerd worden als een IfcMaterialLayerSet bestaande uit 4 lagen (IfcMaterialLayer): gevelsteen, spouw, isolatie en snelbouw.



Figuur 41: IfcMaterialLayerSet (9)

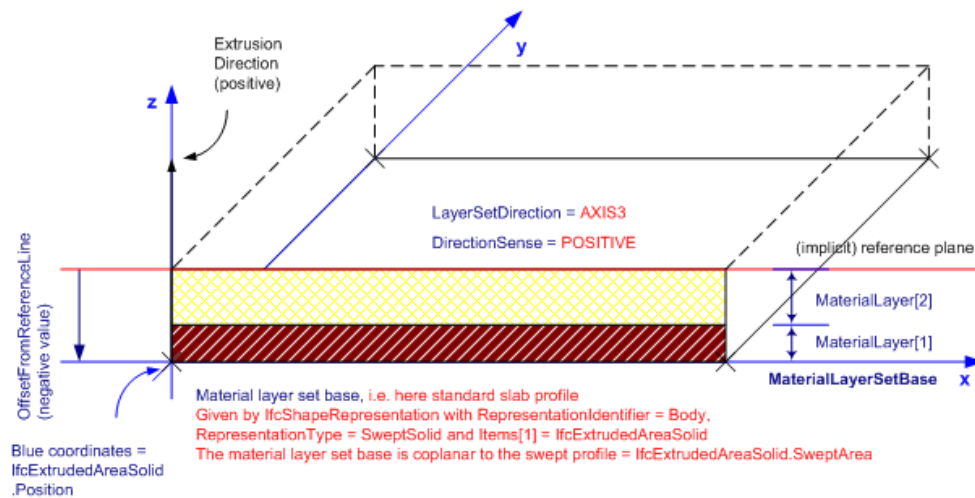
- Een verzameling van materiaallagen (IfcMaterialLayerSetUsage) ingesteld met plaatsbepaling langs de referentie-as of de oppervlakte van het element. De IfcMaterialLayerSetUsage is voornamelijk bedoeld om te worden gebruikt met vlakke bouwelementen met een constante dikte. (muren en vloeren)

Voor een standaard muur (verticale extrusie) zullen de lagen loodrecht op de extrusierichting gelegen zijn. Met de DirectionSense (positief of negatief) kunnen de lagen ten opzichte van de referentie-as van links naar rechts of van rechts naar links opgebouwd worden.



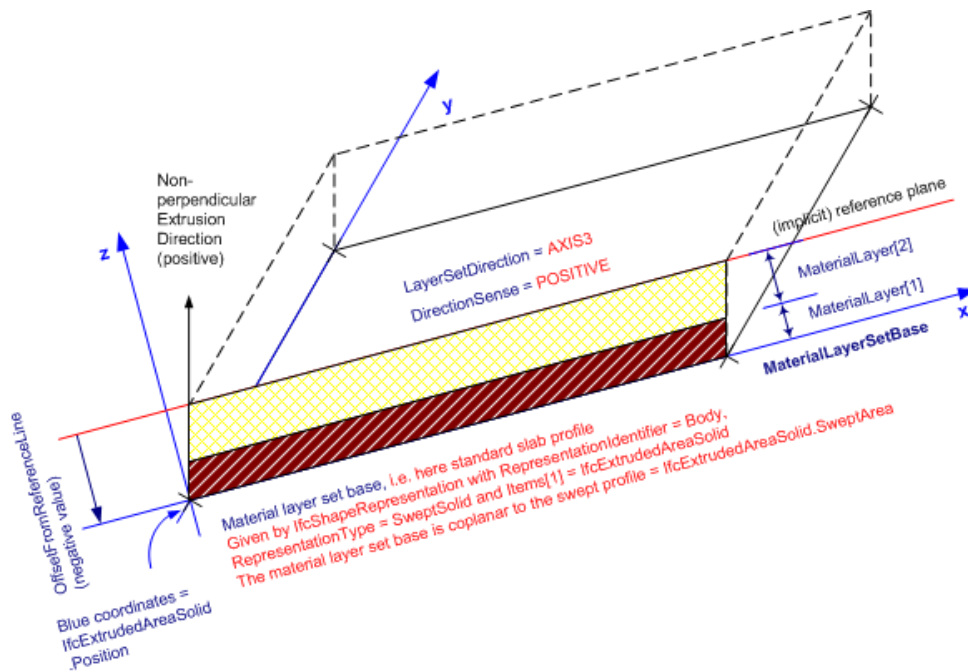
Figuur 42: IfcMaterialLayerSetUsage voor een muur (9)

Bij een vloerplaat liggen de lagen in dezelfde richting als de extrusierichting. Het vlak van een materiaal laag is gelijk aan het geëxtrudeerde profiel. De LayerSetDirection is voor een standaard vloer altijd gelijk aan de z-as.



Figuur 43: IfcMaterialLayerSetUsage voor een vloer (9)

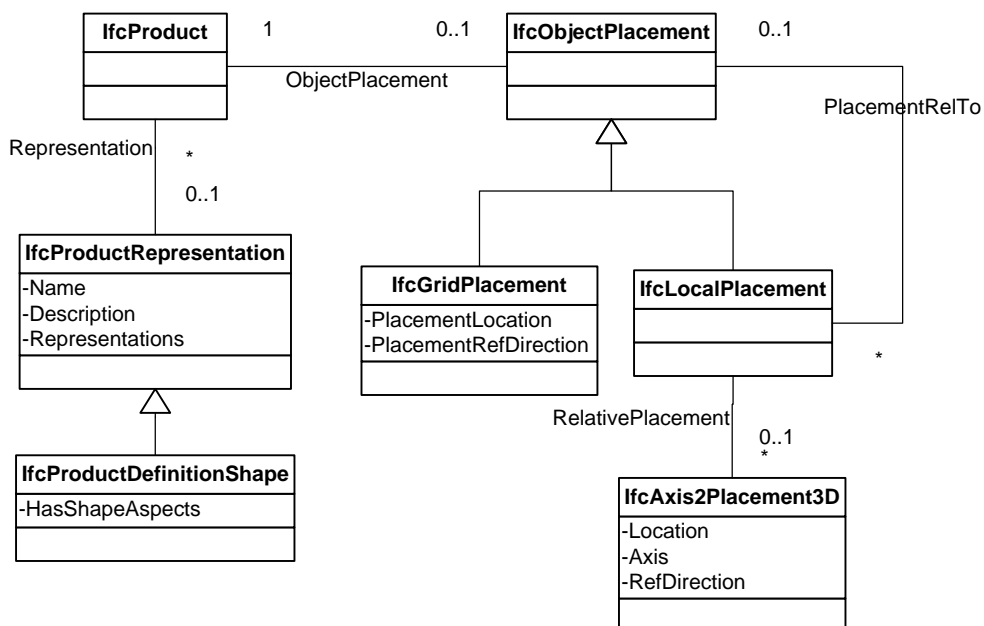
Voor een plaat met een niet-loodrechte extrusie, wordt de laagdikte en de afstand tot de referentielijn altijd loodrecht gemeten. Daarom is de totale dikte van de materiaal lagen niet gelijk aan de extrusiehoogte.



Figuur 44: IfcMaterialLayerSetUsage voor een schuin dak

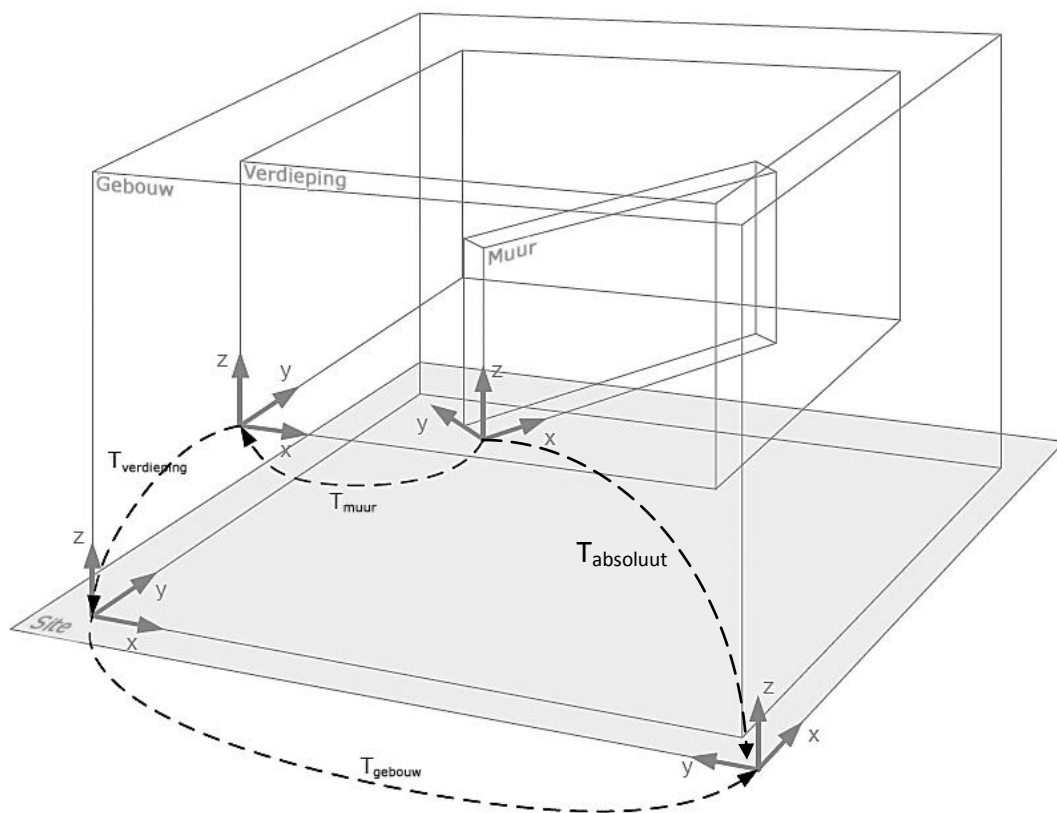
5.4. Plaatsing van objecten

Het ObjectPlacement attribuut beschrijft de plaatsing van het object in de ruimte. De plaatsing kan absoluut (ten opzichte van het globale assenstelsel), relatief (ten opzichte van het assenkruis van een ander object) of beperkt (ten opzichte van assen) zijn. Standaard zal een object relatief geplaatst worden.



Figuur 45: IfcObjectPlacement

De plaatsingen worden door de klasse `IfcLocalPlacement` beschreven. Deze klasse geeft de transformatie weer van de coördinaten van een object ten opzichte van de coördinaten van de container waarin het object zich bevindt. Deze transformatie wordt door de klasse `IfcAxisPlacement3D` beschreven. Als een object relatief geplaatst wordt, dan zal de `PlacementRelTo` verwijzen naar de `IfcLocalPlacement` van het andere object. Om dit te illustreren wordt in Figuur 46 een muur (`IfcWallStandardCase`) weergegeven in een verdieping. Het assenkruis van de muur is gegeven ten opzichte van het assenkruis van de verdieping (T_{muur}). Het assenkruis van de verdieping is op zijn beurt gedeclareerd ten opzichte van het assenkruis van het gebouw ($T_{\text{verdieping}}$), deze is dan weer geplaatst op de site (T_{gebouw}). Wanneer de muur absoluut geplaatst is, dan zou het assenkruis rechtstreeks ten opzichte van het assenkruis van de site opgegeven worden (T_{absoluut}).



Figuur 46: De geometrische hiërarchie via `IfcLocalPlacement` (1)

6. Implementatie

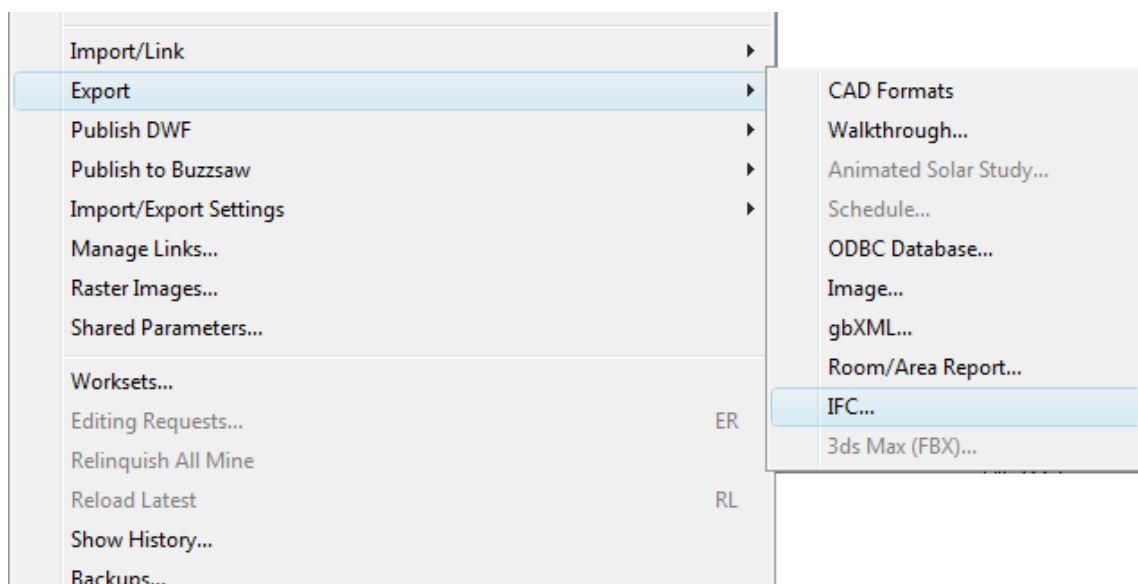
“A program is never finished until the programmer dies.”

Onbekend

Een informatiemodel zal in meerdere stappen in de eigen ontwikkelde applicatie geïmplementeerd worden. Het model wordt eerst in Revit uitgevoerd naar IFC. Het bestand wordt dan geparst en omgezet waarna berekeningen gemaakt kunnen worden met de informatie uit IFC. Er is gekozen om alle toepassingen te ontwikkelen in .NET 3.5¹⁹ en C# 3.0²⁰, die op het moment van schrijven de nieuwste technologieën zijn.

6.1. Exporteren naar IFC

De testcases worden opgemaakt in Autodesk Revit Architecture 2009. De rvt-bestanden worden naar IFC geëxporteerd via File → Export → IFC... Er kan gekozen worden uit IFC2x2, IFC2x3 of IFC BCA ePlan Check. Hier wordt gekozen om naar de laatste versie van IFC, IFC2x3, te exporteren.



Figuur 47: Exporteren van een model naar IFC in Autodesk Revit

6.2. Inlezen van IFC-bestand

Voor het inlezen van een IFC-bestand kon er beroep gedaan worden op de IfcConverter, ontwikkeld aan de UGent door ir. Arch. Ruben Verstraeten. Deze converter vormt een IFC-bestand om naar een XML-bestand met een eigen datastructuur. Deze XML-structuur is

¹⁹ .NET (lees: dotNET) is een applicatieframework ontwikkeld door Microsoft.

²⁰ C# (lees: C Sharp) is een object-georiënteerde programmeertaal ontwikkeld door Microsoft. De syntaxis is gebaseerd op C++.

dus niet gelijk aan de structuur van een ifcXML-bestand. De converter werd ontwikkeld voor de berekening van het K- en E-peil van een woning.

```
<?xml version="1.0"?>
<EPW>
  <Naam>EPW</Naam>
  <Id>EPW_id</Id>
  <Eenheid>mm</Eenheid>
  <KPeilVolumes>
    <Naam>KPeilVolume01</Naam>
    <Id>KPeilVolume01_id</Id>
    <EPeilVolumes>
      <Naam>EPeilVolume01</Naam>
      <Id>EPeilVolume01_id</Id>
      <VentilatieZones>
        <Naam>VentilatieZone01</Naam>
        <Id>VentilatieZone01</Id>
        <EnergieSectoren>
          <Naam>EnergieSector01</Naam>
          <Id>EnergieSector01_id</Id>
          <Spaces>
            .
            .
            .
          </Spaces>
        </EnergieSectoren>
      </VentilatieZones>
    </EPeilVolumes>
  </KPeilVolumes>
</EPW>
```

Code 24: Opbouw van het XML bestand

Na veelvuldige tests is gebleken dat de converter niet voldeed aan de nodige eisen en werd er gekozen voor de ontwikkeling van een eigen parser²¹. De parser is als aparte dll ontwikkeld, zodat die gebruikt kan worden in andere applicaties.

Het SPF-bestand wordt in zijn geheel ingelezen. Daarna wordt een reguliere expressie toegepast. Een reguliere expressie (afgekort tot “regex”, “regex” of RE) is een manier om patronen te beschrijven waarmee een computer tekst kan herkennen. Hieronder is die reguliere expressie te zien.

```
# (?<id>\d+) \s*=\s*(?<entity>[A-Z0-9]+)\s*(?<attributes>.*?);
```

Code 25: De gebruikte reguliere expressie

Door de reguliere expressie wordt onderstaande regel omgezet in:

```
#17=IFCSIUNIT(*, .VOLUMEUNIT., .MILLI., .CUBIC_METRE.);
```

Code 26: Voorbeeldregel uit een IFC bestand

²¹ Een parser is een computerprogramma dat de grammaticale structuur van een invoerbestand volgens een vastgelegde grammatica ontleedt.

- id: 17
- entity: IFCSIUNIT
- attributes: (*, VOLUMEUNIT, MILLI, CUBIC_METRE.)

De tekenreeks “attributes” worden dan verder opgesplitst tot een array van attributen.

Deze gegevens worden in een “Dictionary” opgeslagen. De Dictionary(TKey, TValue) klasse stelt een verzameling van keys en waarden voor.

```
Dictionary<long, Item> IfcDictionary = new Dictionary<long, Item>();
```

Code 27: Voorstelling van de implementatie van een Dictionary in C#

In de eerste kolom worden de id’s opgeslagen. Deze id’s worden dan gebruikt om naar andere entiteiten binnen de dictionary te verwijzen. De tweede kolom bevat de klasse Item. Deze bevat naast de naam (Entity) en de lijst van attributen (Attributes), de eigenlijke klasse (IfcObject) die overeenkomt met de klasse in IFC.

```
internal class Item
{
    public string Entity { get; set; }
    public object IfcObject { get; set; }
    public IList<string> Attributes { get; set; }
}
```

Code 28: De klasse Item

Voor een goed gebruik van de parser zijn de toegangsmodifiers van een viertal functies “public”. Dit wil zeggen dat de functies overal toegankelijk zijn en gebruikt kunnen worden in eigen ontwikkelde applicaties.

6.2.1 Functie OpenModel

Met de functie OpenModel wordt het IFC-bestand ingelezen. Als variabele moet het volledige pad van het bestand worden meegegeven.

```
public static void OpenModel(string path);
```

Code 29: Functie OpenModel

De functie wordt op deze manier opgeroepen:

```
Ifc.Reader.OpenModel(_fileName);
```

Code 30: Het gebruik van de functie OpenModel

6.2.2 Functie GetEntity

Met de functie GetEntity wordt het eerste object uit de Dictionary opgevraagd van het meegegeven type. GetEntity is een generieke functie die voor elk type toegepast kan worden.

```
public static T GetEntity<T>();
```

Code 31: Functie GetEntity

Als voorbeeld wordt een object opgevraagd van het type IfcProject.

```
IfcProject project = Ifc.Reader.GetEntity<IfcProject>();
```

Code 32: Gebruik van de functie GetEntity

6.2.3 Functie GetEntities

Met de functie GetEntities wordt een lijst van objecten uit de Dictionary opgevraagd waarvan het type gelijk is aan het meegegeven type. Net als GetEntity, is GetEntities een generieke functie die voor elk type toegepast kan worden.

```
public static List<T> GetEntities<T>();
```

Code 33: Functie GetEntities

Een “List<T>” is te vergelijken met een array, maar een “List” wordt gebruikt wanneer het niet geweten is hoeveel elementen aan de voorwaarde zullen voldoen.

Onderstaand voorbeeld haalt alle muren uit het model op.

```
List<IfcWall> walls = Ifc.Reader.GetEntities<IfcWall>();
```

Code 34: Gebruik van de functie GetEntities

6.2.4 Functie Version

De functie Version geeft alleen de versie van de parser weer en is voor de verwerking van de gegevens verder nutteloos.

```
public static string Version();
```

Code 35: Functie Version

6.3. Omzetting van klassen

Na het inlezen van het bestand worden de IFC-klassen omgezet naar eigen ontwikkelde klassen. Zo wordt bijvoorbeeld de klasse “IfcDoor” omgezet naar “CDoor”. Deze klassen zorgen voor een verdere verwerking van de informatie, zo berekenen ze het volume, de oppervlakte of de afmetingen van een bouwelement of bepalen of een bouwelement extern of dragend is.

Deze omzetting moet het mogelijk maken om in de toekomst verschillende versies van IFC2x te ondersteunen zonder dat er veel aanpassingswerk nodig is.

Het aanroepen van het model gebeurt door de klasse “CProject”. Deze klasse is tevens het begin (wortel of root) van de boomstructuur (zie Figuur 25 in paragraaf 5.1.1). De knopen worden door “cProject.Nodes” weergegeven.

```
CProject cProject = new CProject();
```

Code 36: Aanroepen van het project

Door het gebruik van een boomstructuur kan men zeer gemakkelijk dingen opvragen. Zo kan men alle bouwelementen op een bepaalde verdieping opvragen. Deze boomstructuur zorgt ook voor een schematische voorstelling van het ganse project.

6.4. Meeteenheden

De meeteenheden die in het IFC-bestand gedefinieerd zijn, worden tijdens de omzetting (zie par. 6.3) omgezet naar de SI eenheden. Wanneer de lengte opgemeten is in millimeter wordt deze omgezet naar meter door het gebruik van onderstaande code.

```
double d = Converter.ToDefault(rectangleProfileDef.XDim.Value, "length");
```

Code 37: Omzetting naar SI eenheid

Om een waarde om te zetten naar een andere waarde moet de functie “Convert” gebruikt worden.

```
Converter.Convert(this.GetThickness(), "millimetre");
```

Code 38: Omzetting naar andere eenheid

De functie “ConvertStr” zet de waarde om naar een andere waarde en zet er de eenheid achter. Bvb. 125 mm

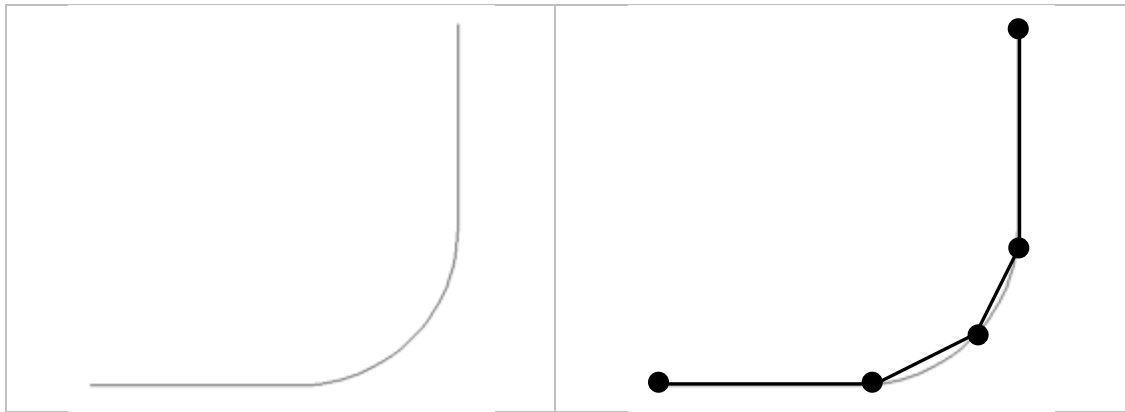
```
Converter.Convert(this.GetThickness(), "millimetre");
```

Code 39: Omzetting naar andere eenheid in stringformaat

6.5. Geometrie

Wegens tijdsgebrek en de moeilijkheidsgraad van implementatie is maar een deel van de geometrische voorstelling uit paragraaf 5.2 ingebouwd. Er is enkel mogelijk om het volume te berekenen van elementen die opgebouwd kunnen worden door middel van extrusie. Dit is de meest eenvoudigste geometrische voorstelling die mogelijk is.

Voor de berekening worden alle grondvlakken omgezet naar polygonen. Wanneer grondvlakken afrondingen bevatten worden ook deze afrondingen omgezet naar lijnstukken. Dit geeft echter een benadering van de correcte oppervlakte van een grondvlak. Om een betere benadering te krijgen dient er nog een verdere verfijning te gebeuren. Zo kan de opdeling van de afronding in lijnstukken afhankelijk gemaakt worden van de lengte van de cirkelboog en de straal.



Figuur 48: Omzetting afronding naar polygoon

De berekening van de oppervlakte van een polygoon gebeurt met onderstaande formule.

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i)$$

Vergelijking 1: Berekening oppervlakte van polygoon

De oppervlakte wordt dan geëxtrudeerd langsheen een pad. De lengte van het pad wordt berekend om zo tot het volume te komen. In eerste instantie wordt het volume van elk bouwelement berekend. Hieruit kan dan de oppervlakte en afmetingen van het element bepaald worden.

6.6. Uitlezing van gegevens

Voor de uitlezing van gegevens kan er gekozen worden om op te slaan naar Excel of PDF.

6.6.1 Excel

Voor de uitlezing naar Excel is er geopteerd om een eigen exporter te ontwikkelen. Als bestandsformaat is gekozen voor Spreadsheet XML 2003. Dit is een bestandsformaat ontwikkeld door Microsoft voor de inlezing van gegevens in Microsoft Excel 2003. Spreadsheet XML is een open bron bestandsformaat met de structuur van een XML bestand.

Voordelen:

- Snel en eenvoudig
- Open bron
- XML technologie
- Microsoft Office niet nodig voor aanmaken bestand

Nadelen:

- Het gebruik van afbeeldingen en grafieken zijn niet mogelijk

Hoewel Office Open XML (OOXML) net als Spreadsheet XML een open bron formaat is en gebruik maakt van XML voor de weergave van gegevens, is hier niet voor gekozen. De opbouw van een OOXML-bestand is ingewikkelder en een OOXML is enkel te openen in oudere versies van Microsoft Office met behulp van een converter.

Net als de IFC-parser is de Excel-converter in een aparte dll gegoten, zodat andere applicaties hiervan gebruik kunnen maken.

Voorbeeld voor de aanmaak van een eenvoudige tabel:

```
using Exporter.Excel;

public class Form1
{
    private void Button1_Click(System.Object sender, System.EventArgs e)
    {
        using (SaveFileDialog saveFileDialog = new SaveFileDialog()) {
            saveFileDialog.Filter = "XML-werkblad 2003 (*.xml)|*.xml";
            if (DialogResult.OK == saveFileDialog.ShowDialog()) {

                // Werkboek aanmaken
                Workbook book = new Workbook();

                // Werkblad toevoegen
                Worksheet sheet = book.Worksheets.Add("Blad 1");

                Row row;
                for (i = 0; i <= 10; i++) {
                    row = sheet.Table.Rows.Add();
                    row.Cells.Add("waarde 1");
                    row.Cells.Add("waarde 1");
                    row.Cells.Add("waarde 1");
                }

                // Excelbestand opslaan
                book.Save(saveFileDialog.FileName);
            }
        }
    }
}
```

Niet alle mogelijkheden van Spreadsheet XML werden ingebouwd wegens tijdsgebrek en voor deze thesis niet van toepassing. Voor een meer uitgebreide uitleg wordt er naar de handleiding verwezen die terug te vinden is op de CD-rom.

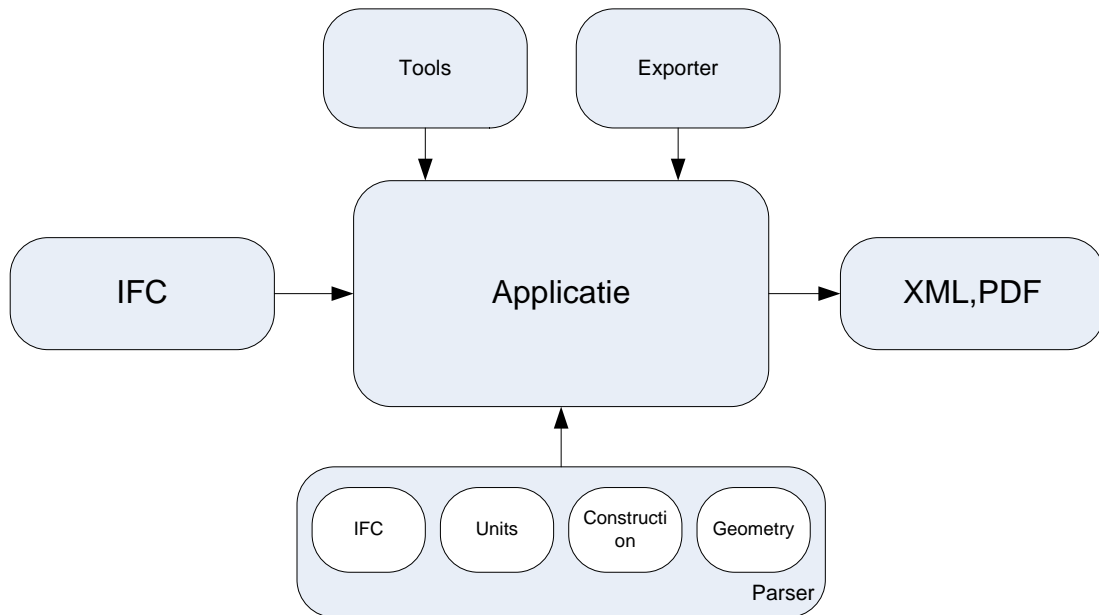
6.6.2 PDF

Voor het opslaan naar PDF is er gebruik gemaakt van iText#²². iText# is ontwikkeld uit de open bron iText voor JAVA en volledig in C# ontwikkeld. iText is een bibliotheek om dynamische PDF documenten te genereren.

²² <http://itextsharp.sourceforge.net/>

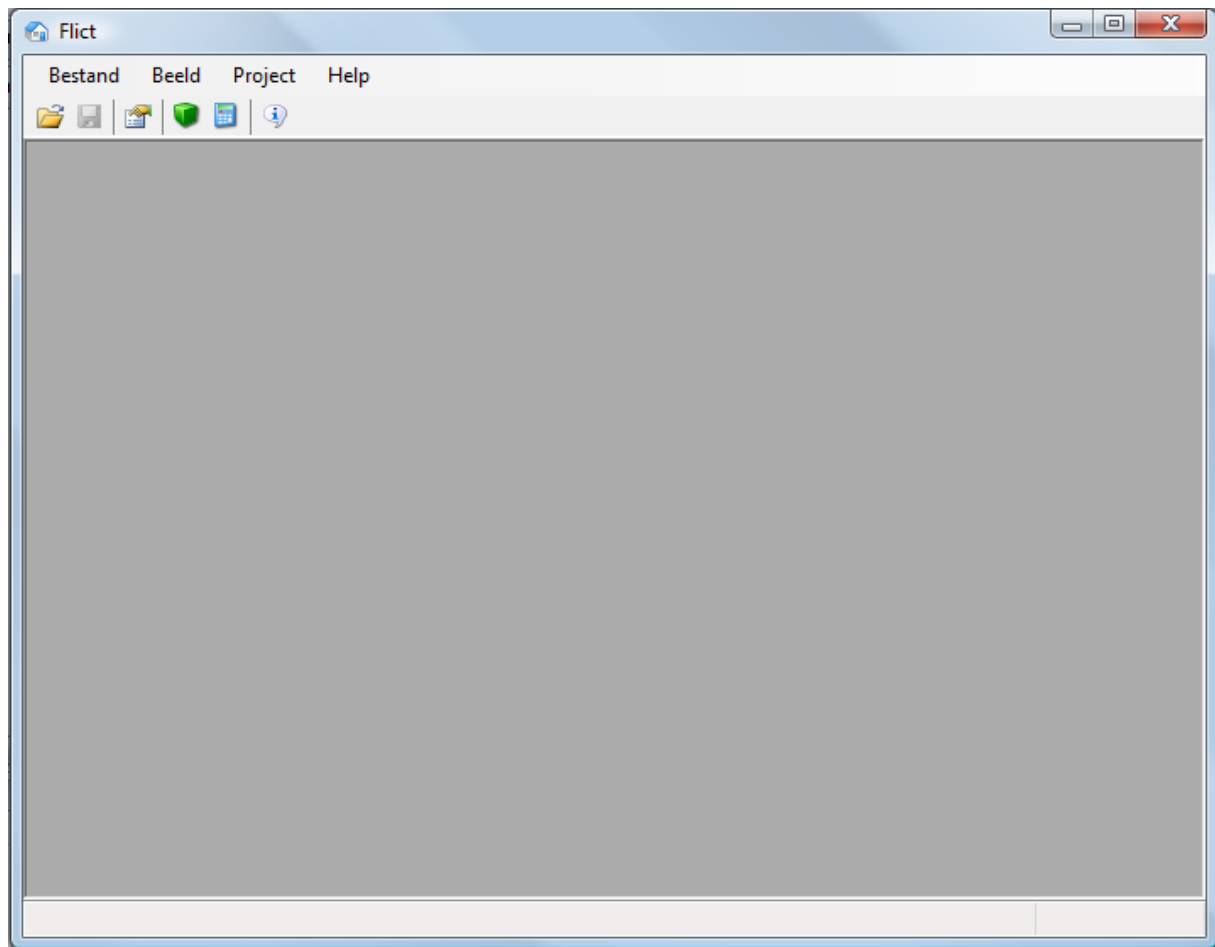
6.7. Grafische User Interface

Een grafische user interface (GUI) werd ontworpen die bovenstaande elementen combineert tot een bruikbare applicatie.



Figuur 49: Voorstelling applicatie

De applicatie heeft de naam FLICT meegekregen. De naam is een samenraapsel van de letters IFC en mijn initialen TL. De GUI bevat drie belangrijke vensters: de Projecteigenschappen, de Viewer en de Meetstaat.



Figuur 50: GUI van FLICT

6.7.1 Project-eigenschappen

Het venster bevat drie tabbladen met informatie over het project. Het eerste tabblad geeft algemene informatie, zoals de naam en beschrijving van het project, de gebruiker en eventueel de bouwfase.

Het tweede tabblad bevat de gebruikte eenheden waarin het project is opgemeten.

Units	
Units	Format
length	centimetre [cm]
area	square centimetre [cm ²]
volume	cubic centimetre [cm ³]
time	second [s]
currency	EUR [€]
mass	gram [g]
thermodynamictemperature	kelvin [K]
planeangle	degree [°]

Figuur 51: Venster met de eenheden

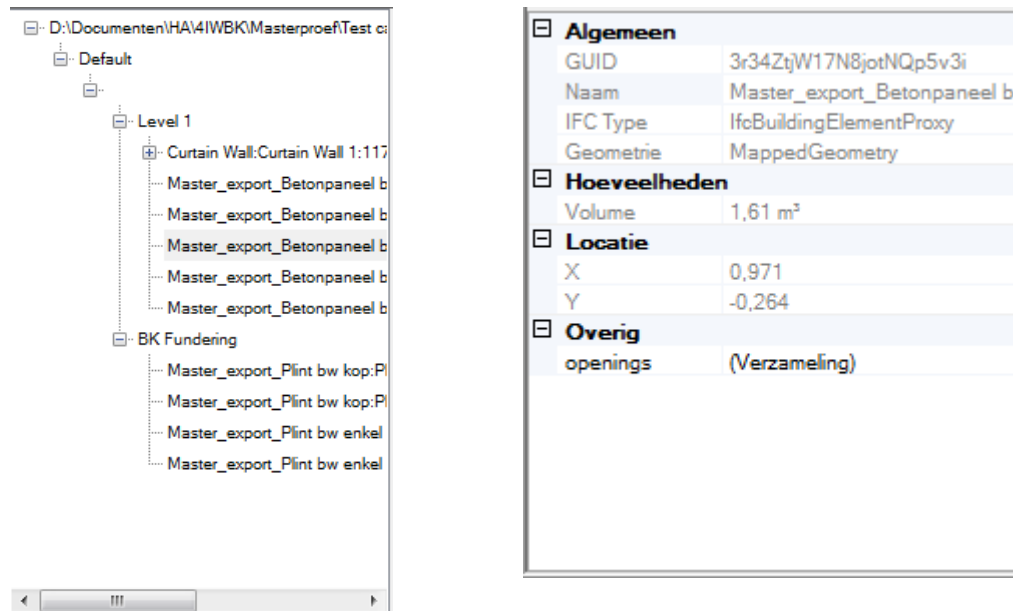
Het derde tabblad geeft een opsomming van de gebruikte materialen en materiaallagen.

Materials		
Layers Materialen		
Floor: Insulated floor 310mm		
Totale Dikte <input type="text" value="310"/> mm		
	Material	Thickness [mm]
▶ 1	Ceramic Tile	10
2	Insulation / Thermal Barriers - Rigid insulation	80
3	Concrete - Sand/Cement Screed	50
4	Concrete - Cast In Situ	170

Figuur 52: Venster met de materiaallagen

6.7.2 Viewer

Dit venster bevat de boomstructuur van het project en geeft de eigenschappen van elk bouwelement. Wegens tijdsgebrek is er geen visualisatie van het project mogelijk.



Figuur 53: Weergave boomstructuur(links) en informatie van een element (rechts)

6.7.3 Meetstaat

Dit venster bevat de functies voor de berekeningen van de stuklijst en algemene meetstaat.

Betonpaneel bw																												
	Lengtl	Offset	Work Plane	Lengtl	Start Relea	Start Fx	Start Fy	Start Fz	Start Mx	Start My	Start Mz	End Relea	End Fx	End Fy	End Fz	End Mx	End My	End Mz	Lengtl	Volum	Mark	Hoogt	Dikte	Descr	Assem Code	Assem Descr	Aaljn	Aantal
1	900	0	Mas...	980	1	False	False	False	False	True	True	1	False	False	False	True	True	True	575...	B1	4200	140	Bet...			70	1	
2	400...	4500	Lev...	399...	1	False	False	False	False	True	True	1	False	False	False	True	True	True	161...	B2	2700	150	Bet...			75	1	
3	7500	4500	Lev...	7500	1	False	False	False	False	True	True	1	False	False	False	True	True	True	283...	B3	2700	140	Bet...			70	1	
4	650...	4500	Lev...	649...	1	False	False	False	False	True	True	1	False	False	False	True	True	True	262...	B4	2700	150	Bet...			75	1	

cellenbetonpaneel										
	Area	Width	Height	Mark	Thickness	Offset	Description	Assembly Description	Assembly Code	Aantal
1	1800000	3000	600	type1	100	100	cellenbetonpaneel			6
2	4500000	7500	600	type2	100	100	cellenbetonpaneel			6
3	2994000	4990	600	type3	100	100	cellenbetonpaneel			6
4	1800000	3000	600	type4	100	100	cellenbetonpaneel			1

Plint																												
	Lengtl	Offset	Work Plane	Lengtl	Start Relea	Start Fx	Start Fy	Start Fz	Start Mx	Start My	Start Mz	End Relea	End Fx	End Fy	End Fz	End Mx	End My	End Mz	Lengtl	Volum	Mark	Hoogt	Dikte	Descr	Assem Code	Assem Descr	Aaljn	Aantal
1	400	0	Lev...	399...	1	False	False	False	False	True	True	1	False	False	False	True	True	True	614...	P1	1100	140	Plint			70	1	
2	7500	0	Lev...	7480	1	False	False	False	False	True	True	1	False	False	False	True	True	True	115...	P2	1100	140	Plint			70	1	
3	5000	0	Lev...	4980	1	False	False	False	False	True	True	1	False	False	False	True	True	True	765...	P3	1100	140	Plint			70	1	

Figuur 54: Stuklijst

Nr	Code	Beschrijving	Markttype	Meeteenheid	Hoeveelheid	Eenheidsprijs	Totaal
		Default Wall	VH	m3	5,78		
		Ceramic Tile	VH	m3	3,92		
		Insulation / Thermal Barriers - Rigid insulation	VH	m3	54,9		
		Concrete - Sand/Cement Screed	VH	m2	392,16		
		Concrete - Cast In Situ	VH	m3	98,04		
		Roofing - Asphalt	VH	m3	3,92		
		Masonry - Brick	VH	m3	26,66		
		Air Barrier - Air Infiltration Barrier	VH	m3	8,89		
		Insulation / Thermal Barriers - Cavity Fill	VH	m3	17,78		
		Masonry - Concrete Block	VH	m3	41,48		
		Plasterboard	VH	m3	3,56		

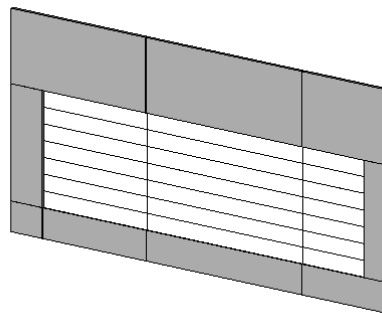
Figuur 55: Meetstaat

7. Tests

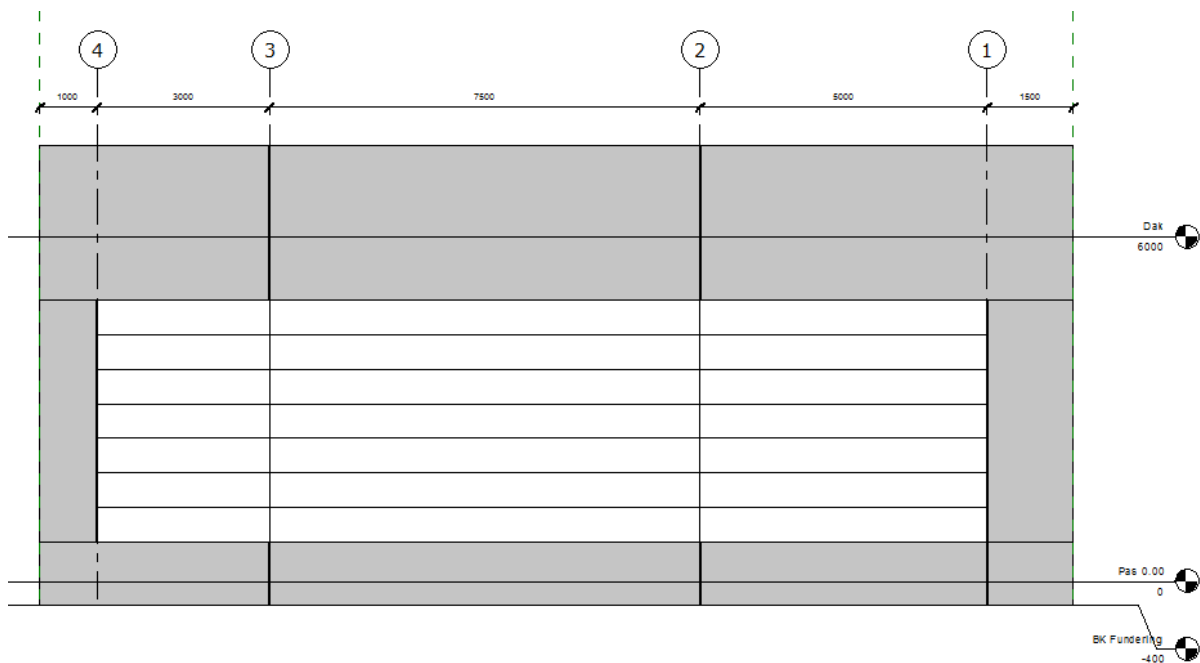
7.1. TestCase 1

7.1.1 Beschrijving

De eerste testcase is aangereikt door de firma Bolckmans NV en bestaat uit een brandwand opgebouwd uit cellenbetonpanelen en betonnen elementen. De cellenbetonpanelen zijn weergegeven als IfcPlates (21) die gegroepeerd zijn in een IfcCurtainWall. De betonnen elementen bevatten zowel de plinten als panelen en worden door IfcBuildingElementProxy's weergegeven.



Figuur 56: 3D weergave van brandwand



Figuur 57: Vooraanzicht brandwand

7.1.2 Probleemstelling

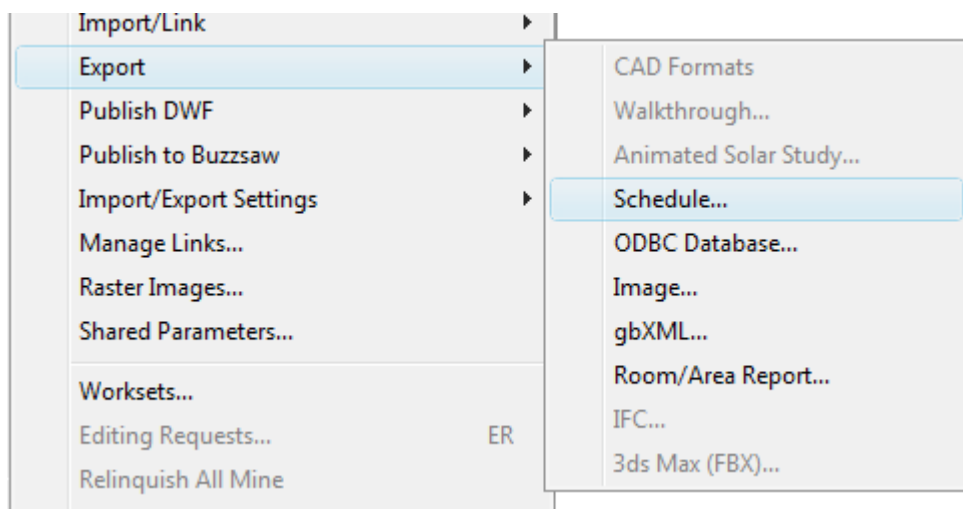
In Revit is het mogelijk om zogenaamde Schedules of Quantities Views te genereren. Dit is een tabelweergave van het getekend model, waarbij de gegevens afhangen van de eigenschappen van bepaalde componenten. De tabellen worden automatisch geüpdate wanneer er wijzigingen gebeuren aan het model. Deze schedules kunnen, net zoals planzichten, op een Drawing Sheet geplaatst worden.

Betonpanelen					
Mark	Lengte	Hoogte	Dikte	Description	Count
B1	980	4200	140	Betonpaneel bw	1
B5	1490	4200	140	Betonpaneel bw	1
B2	3990	2700	150	Betonpaneel bw	1
B4	6490	2700	150	Betonpaneel bw	1
B3	7500	2700	140	Betonpaneel bw	1

Figuur 58: Schedule in Revit

Dergelijke tabellen zijn gemakkelijk op te bouwen aan de hand van queries²³. Zo kan men kiezen welke eigenschappen weergegeven worden en hoe er gesorteerd en gegroepeerd moet worden.

De tabellen kunnen ook uitgevoerd worden naar een tekstbestand om verder gebruikt te worden in klassieke meetstaten en bestekken.



Figuur 59: Export van een schedule in Revit

Het is niet altijd duidelijk welke tabellen geüpdate worden en welke niet. Men moet dus alle tabellen overlopen en terug uitvoeren naar een tekstbestand. Dit proces neemt veel tijd in beslag en is niet nauwkeurig.

²³ Een query is een opdracht die aan een databank wordt gegeven om een bepaalde actie uit te voeren.

7.1.3 Uitwerking

Er moet dus een oplossing gezocht worden om dit probleem uit de weg te gaan. Mogelijke oplossingen:

- ODBC: Revit biedt naast de uitvoer van het informatiemodel naar IFC ook een uitvoer via ODBC naar een databank (Access of Excel). ODBC staat voor Open DataBase Connectivity en is ontwikkeld om elk programma met een databank te laten spreken, onafhankelijk van het type. Revit gebruikt een naamgeving voor de tabellen die niet dezelfde is als in IFC. Omdat het teveel tijd zou vergen om deze weergave onder de knie te krijgen, valt deze optie weg.
- VSTA staat voor Visual Studio for Applications en is de opvolger van Visual Basic for Applications (VBA). Men kan behalve Visual Basic ook de andere .NET-talen, zoals C#, gebruiken. Revit biedt met VSTA een eigen programmeeromgeving aan. Omdat de kennis over de VSTA-module nihil is en Revit de mogelijkheden van de module beperkt, is ook deze denkpiste achterwege gelaten.
- Eigen methode: ontwikkeling van een eigen methode om de nodige informatie uit IFC te halen

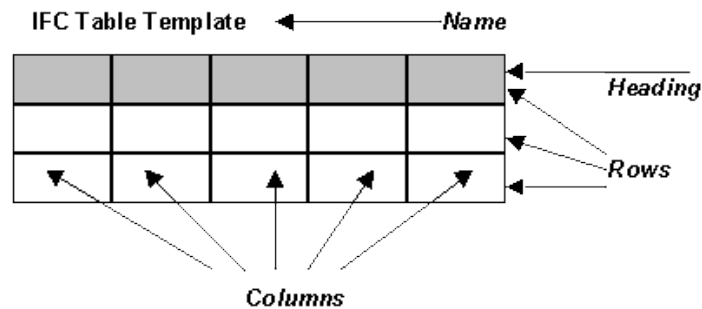
De schedules die in Revit aangemaakt zijn worden niet geëxporteerd naar IFC, noch in tabelvorm noch in de vorm van een query. De schedules moeten dus terug opgebouwd worden met de informatie uit IFC. Men gaat dubbel werk leveren, eerst de schedules in Revit opbouwen en daarna nog eens in de applicatie.

Er bestaat echter een mogelijkheid om in IFC tabellen weer te geven. Met behulp van de klassen `IfcTable` en `IfcTableRow` kan een tabel gedefinieerd worden.

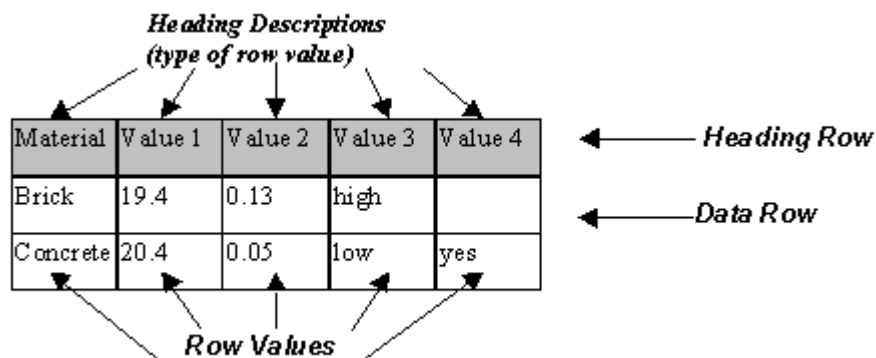
```
#120 = IFCTABLEROW ((IFCDESCRIPTIVEMEASURE ('Fill Name'),  
IFCDESCRIPTIVEMEASURE ('Pattern8')), .T.);  
#121 = IFCTABLEROW ((IFCDESCRIPTIVEMEASURE ('common brick'), IFCINTEGER (-  
120)), .F.);  
#122 = IFCTABLE ('dDSG9s82/e;4Gcxh<Uo<', 'Fill-Material Table', (#120,  
#121));
```

Code 40: Tabel in IFC

Op het eerste zicht lijkt het interessant om schedules als `IfcTables` weer te geven, maar een `IfcTable` heeft één groot nadeel. In de cellen kunnen enkel `IfcValues` ingegeven worden. Dit maakt de er geen enkel verband ontstaat tussen de waarde in een cel en vanwaar de waarde komt. Bij een aanpassing in het tekenmodel zal het dus niet mogelijk zijn de waardes in de tabel up te daten.



Figuur 60: Weergave van IfcTable



Figuur 61: Weergave van IfcTableRow

Het is echter niet interessant om schedules weer te geven als IfcTables. In de cellen kan enkel een IfcValue ingegeven worden en zo is er geen enkel verband meer waar de waarden voor staan en kunnen dus ook niet meer aangepast worden als er iets in het tekenmodel veranderd.

De query zal dus handmatig terug moeten ingegeven worden. Hiervoor is gebruik gemaakt van LINQ. Dit is een nieuwe technologie in C# 3.0 en het biedt een werkwijze aan voor een meer uniforme omgang met gegevens uit heel verschillende systemen, zoals databases, webservices, XML of arrays. Al deze verschillende soorten gegevens kunnen hiermee opgevraagd worden met een op SQL-lijkende query. Het gebruik van LINQ maakt de programmacode niet sneller, maar wel overzichtelijker en beter leesbaar.

De functie `GetProperty` gaat een welbepaalde eigenschap van een element gaan opvragen. Op de teruggestuurde van de functie wordt dan gesorteerd en gegroepeerd. Voor de eenvoud werkt deze query enkel bij deze testcase omdat de query in de code al is opgebouwd. In principe kan deze query dynamisch opgebouwd kunnen worden aan de hand van een invoerscherm gelijkend op het invoerscherm in Revit.

```
list.OrderBy(l => l.GetProperty("Description")).GroupBy(l =>
    l.GetProperty("Description")).Select(
    d => new
    {
        Description = d.Key,
        Marks =
        d.OrderBy(l => l.GetProperty("Mark")).GroupBy(l
            => l.GetProperty("Mark")).Select(
            m => new { Mark = m.Key, Elements = m })
    });
```

Code 41: Gebruikte LINQ query

Bovenstaande query geeft onderstaande oplossing. De elementen worden gegroepeerd volgens beschrijving (cellenbetonpaneel, plint of betonpaneel). In de tabel worden alle eigenschappen van het element opgesomd. Het is aan de gebruiker om enkel de nuttige eigenschappen aan te vinken door middel van het keuzevenster (zie Figuur 63).

Betonpaneel bw

	Lengt	Offset	Work Plane	Lengt	Start Relea	Start Fx	Start Fy	Start Fz	Start Mx	Start My	Start Mz	End Relea	End Fx	End Fy	End Fz	End Mx	End My	End Mz	Lengt	Volum	Mark	Hoogt	Dikte	Descr	Assen Code	Assen Descr	Aaljn	Aantal
1	900	0	Mas...	980	1	False	False	False	False	True	True	1	False	False	False	True	True	True	575...	B1	4200	140	Bet...			70	1	
2	400	4500	Lev...	399	1	False	False	False	False	True	True	1	False	False	False	True	True	True	161...	B2	2700	150	Bet...			75	1	
3	7500	4500	Lev...	7500	1	False	False	False	False	True	True	1	False	False	False	True	True	True	283...	B3	2700	140	Bet...			70	1	
4	650	4500	Lev...	649	1	False	False	False	False	True	True	1	False	False	False	True	True	True	262...	B4	2700	150	Bet...			75	1	

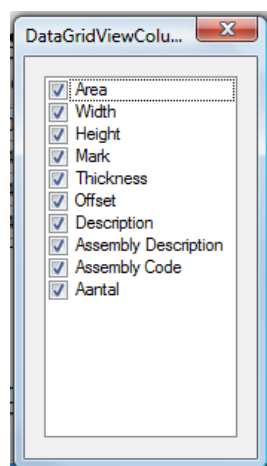
cellenbetonpaneel

	Area	Width	Height	Mark	Thickness	Offset	Description	Assembly Description	Assembly Code	Aantal
1	1800000	3000	600	type1	100	100	cellenbetonpaneel			6
2	4500000	7500	600	type2	100	100	cellenbetonpaneel			6
3	2994000	4990	600	type3	100	100	cellenbetonpaneel			6
4	1800000	3000	600	type4	100	100	cellenbetonpaneel			1

Plint

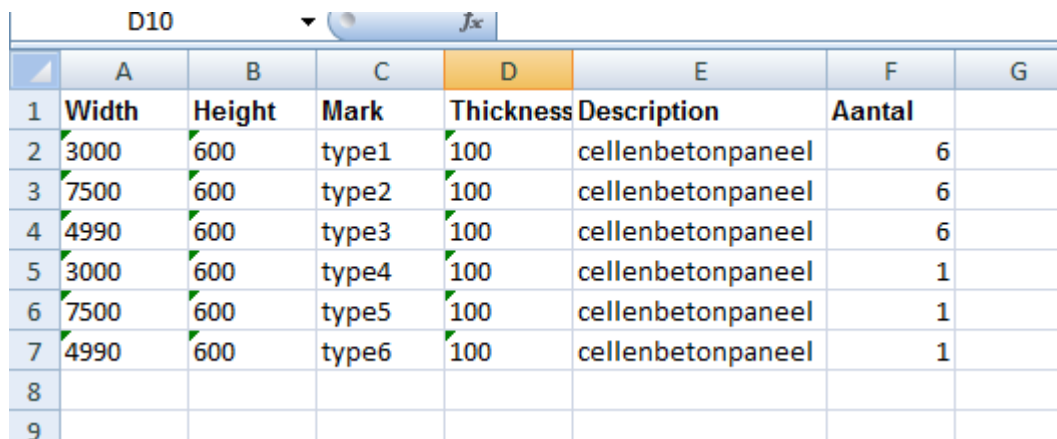
	Lengt	Offset	Work Plane	Lengt	Start Relea	Start Fx	Start Fy	Start Fz	Start Mx	Start My	Start Mz	End Relea	End Fx	End Fy	End Fz	End Mx	End My	End Mz	Lengt	Volum	Mark	Hoogt	Dikte	Descr	Assen Code	Assen Descr	Aaljn	Aantal
1	400	0	Lev...	399	1	False	False	False	False	True	True	1	False	False	False	True	True	True	614...	P1	1100	140	Plint			70	1	
2	7500	0	Lev...	7480	1	False	False	False	False	True	True	1	False	False	False	True	True	True	115...	P2	1100	140	Plint			70	1	
3	5000	0	Lev...	4980	1	False	False	False	False	True	True	1	False	False	False	True	True	True	765...	P3	1100	140	Plint			70	1	

Figuur 62: Resultaat in FLICHT



Figuur 63: Keuzevenster

Elke tabel kan nu opgeslagen worden in XML-formaat voor inlezing in Excel en in PDF-formaat.



	A	B	C	D	E	F	G
1	Width	Height	Mark	Thickness	Description	Aantal	
2	3000	600	type1	100	cellenbetonpaneel	6	
3	7500	600	type2	100	cellenbetonpaneel	6	
4	4990	600	type3	100	cellenbetonpaneel	6	
5	3000	600	type4	100	cellenbetonpaneel	1	
6	7500	600	type5	100	cellenbetonpaneel	1	
7	4990	600	type6	100	cellenbetonpaneel	1	
8							
9							

Figuur 64: Weergave tabel Cellenbetonpaneel in Excel

De schedule kunnen op per bouwfase gegenereerd worden in Revit. Maar omdat de bouwfase niet naar IFC geëxporteerd wordt, is dit in de applicatie niet mogelijk. Een mogelijke oplossing is om de bouwfase via een extra gedefinieerde eigenschap weer te geven zodat deze zeker naar IFC wordt geëxporteerd, maar de optie werd niet verder uitgewerkt.

7.2. Testcase 2

7.2.1 Beschrijving

In samenwerking met Yves Van Looy is er een gemeenschappelijke testcase ontwikkeld. De testcase bestaat uit een eenvoudige woning met twee verdiepingen en drie ruimtes.

In de woning zijn twee types van muren. De buitenmuren zijn opgebouwd uit 5 lagen:

- Masonry – Brick
- Air Barrier – Air Infiltration Barrier
- Insulation / Thermal Barriers – Cavity Fill
- Masonry – Concrete Block
- Plasterboard

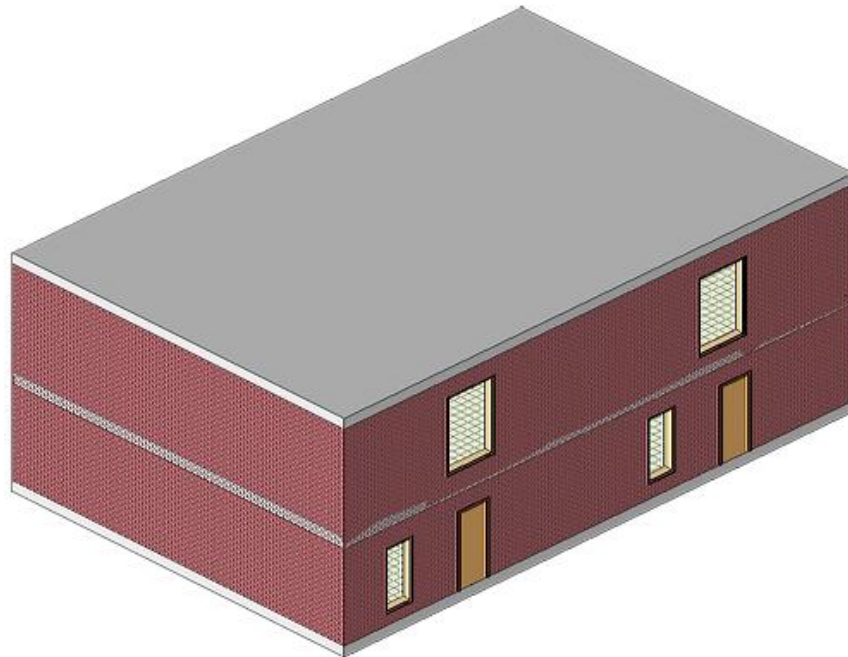
De binnenmuur bestaat uit één enkele laag.

Het dak is opgebouwd uit 3 lagen:

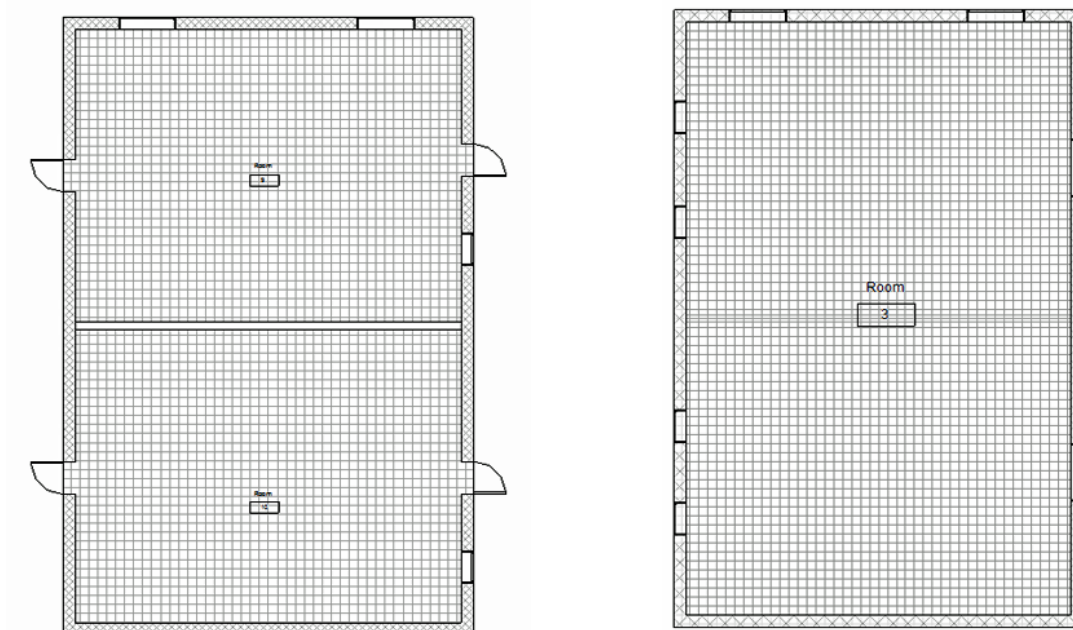
- Roofing – Asphalt
- Insulation / Thermal Barriers – Rigid insulation
- Concrete – Cast in Situ

De vloeren bestaan uit 4 lagen:

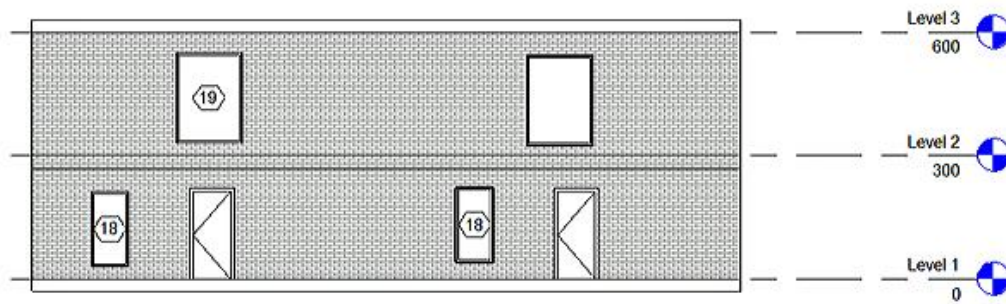
- Ceramic Tile
- Insulation / Thermal Barriers – Rigid insulation
- Concrete – Sand/Cement Screed
- Concrete – Cast in Situ



Figuur 65: 3D weergave woning



Figuur 66: Weergaves van grondplan van gelijkvloers (links) en eerste verdieping (rechts)



Figuur 67: Vooraanzicht woning

7.2.2 Probleemstelling

De bedoeling van deze testcase is om een algemene meetstaat te genereren. Daarin moeten alle gebruikte materialen en vensters en deuren staan.

7.2.3 Uitwerking

Allereerst wordt de boomstructuur van het model doorlopen en alle bouwelementen worden in een aparte lijst opgeslagen. Omdat we hier werken met elementen die opgevraagd zijn met de inverse relatie `ContainsElements` van een ruimtelijke structuur, bevat deze lijst geen dubbele bouwelementen. Wanneer we de elementen zouden opvragen met de inverse functie `BoundedBy` van een ruimte, deze functie geeft de grenzen van een ruimte, dan bestaat de kans dat sommige elementen dubbel in de lijst voorkomen.

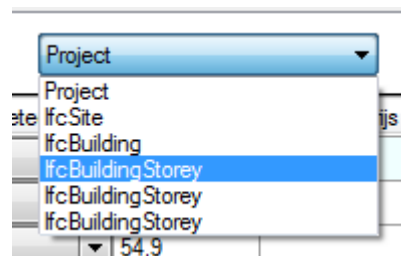
Van elk bouwelement wordt dan zijn materialen opgevraagd. Om het volume van elke materiaallaag te bekomen, wordt het ganse volume van de muur vermenigvuldigd met de laagdikte en gedeeld door de volledige dikte van de muur. Materialen met dezelfde eigenschappen worden bij elkaar opgeteld. Het volume van een bouwelement is steeds het nettovolume, dat wil zeggen dat openingen niet meegeteld worden.

De hoeveelheden worden standaard in de meetstaat weergegeven in kubieke meter, maar kunnen eenvoudig naar vierkante meter omgezet worden. Bij staal zal de hoeveelheid enkel gegeven worden in kilogram. Deuren en vensters worden in stuks weergegeven.

Nr	Code	Beschrijving	Markttype	Meeteenheid	Hoeveelheid	Eenheidsprijs	Totaal
		Default Wall	VH	m3	5,78		
		Ceramic Tile	VH	m3	3,92		
		Insulation / Thermal Barriers - Rigid insulation	VH	m3	54,9		
		Concrete - Sand/Cement Screed	VH	m2	392,16		
		Concrete - Cast In Situ	VH	m3	98,04		
		Roofing - Asphalt	VH	m3	3,92		
		Masonry - Brick	VH	m3	26,66		
		Air Barrier - Air Infiltration Barier	VH	m3	8,89		
		Insulation / Thermal Barriers - Cavity Fill	VH	m3	17,78		
		Masonry - Concrete Block	VH	m3	41,48		
		Plasterboard	VH	m3	3,56		

Figuur 68: Meetstaat in FLICT

Men kan de meetstaat ook laten genereren voor elke ruimtelijke structuur, per gebouw, per verdieping of per site. Zo krijgt men bijvoorbeeld enkel een opsomming van gebruikte materialen voor de eerste verdieping.



Figuur 69: Keuzelijst van ruimtelijke structuren

De gegevens kunnen naar Excel uitgevoerd worden en zo gebruikt worden in calculatieprogramma's.

	A	B	C	D	E	F	G	H	I
1	Nr	Code	Beschrijving	Markttype	Meeteenheid	Hoeveelheid	Eenheidsprijs	Totaal	
2			Default Wall	VH	m3	5,78			
3			Ceramic Tile	VH	m3	3,92			
4			Insulation / Thermal Barriers - Rigid insulation	VH	m3	54,9			
5			Concrete - Sand/Cement Screed	VH	m3	19,61			
6			Concrete - Cast In Situ	VH	m3	98,04			
7			Roofing - Asphalt	VH	m3	3,92			
8			Masonry - Brick	VH	m3	26,66			
9			Air Barrier - Air Infiltration Barrier	VH	m3	8,89			
10			Insulation / Thermal Barriers - Cavity Fill	VH	m3	17,78			
11			Masonry - Concrete Block	VH	m3	41,48			
12			Plasterboard	VH	m3	3,56			
13									
14									

Figuur 70: Meetstaat in Excel

Voor een eerste zeer eenvoudige raming van de kostprijs zijn twee extra kolommen toegevoegd, eenheidsprijs en totaal. In de kolom eenheidsprijs kan een prijs met meeteenheid ingegeven worden en automatisch wordt deze eenheidsprijs vermenigvuldigd met de hoeveelheid tot een totaal.

Een verdere verfijning kan nog toegepast worden door de bouwmaterialen te ordenen volgens categorie. Zo kan men de materialen "Masonry – Concrete Block" en "Masonry – Brick" groeperen tot een groep "Masonry". Dit kan gedaan worden volgens naamgeving. Zo kan men alle materialen overlopen en de materialen beginnend met Masonry groeperen. Dit is niet altijd mogelijk omdat de naamgeving door de gebruiker zelf kan opgegeven worden. Een oplossing hiervoor zal te vinden in de volgende versie van IFC2x, IFC2x4. De entiteit krijgt een extra parameter, namelijk category. Men kan dan gaan groeperen op category (zie par. 4.7).

8. Besluit

Eerste en vooral moet er gezegd worden dat IFC geen gemakkelijke materie is. De structuur is specifiek gebouwd op overerving en relaties tussen entiteiten. Dit maakt het voor een beginneling soms onoverzichtelijk. Men kan als het ware door de bomen het bos niet meer zien. Gelukkig is er genoeg informatie te vinden, zodat men toch een beetje wegwijs geraakt in dit kluwen.

Tijdens de thesis is steeds gebleken dat de één op één relatie tussen software applicaties, wat toch het doel is van IFC, allesbehalve 100% werkt. De uitwisseling is in sommige gevallen zelfs ondermaats. Wat een grote hoeveelheid van informatieverlies met zich meebrengt. Men kan dan de vraag stellen of deze software applicaties wel de certificering van IAI verdienen. De waarde van de certificering verdwijnt hierdoor in het niets. Maar als men de certificering te streng maakt, zal geen enkele applicatie slagen en komt het IFC-bestandsformaat in het gedrang.

Er moet wel gezegd worden dat IFC nog steeds groeit en dat IFC waarschijnlijk nooit volledig zal zijn. IFC is momenteel nog niet volwassen genoeg om te kunnen opboksen tegen andere bestandsformaten van grote softwareproducenten. Er wordt wel eens gezegd dat IFC ten dode is opgeschreven, maar hoe meer mensen zich gaan bezighouden met de ontwikkeling en implementatie, hoe groter het aandeel van IFC wordt binnen de softwareapplicaties. Waarschijnlijk zal de SPF-weergave van IFC op termijn verdwijnen en plaatsmaken voor modernere uitwisselingsformaten, ifcXML en de zogeheten BIM-servers.

De ontwikkeling van een eigen parser voor IFC was misschien niet de beste oplossing, omdat het veel te veel tijd in beslag genomen heeft. Maar dit heeft er wel voor gezorgd dat ik de werking van IFC beter onder de knie kreeg. Een ander tijdrovend proces was het constant aanpassen en verbeteren van de reeds ontwikkelde code aan de hand van testbestanden. Dit komt omdat IFC een te grote vrijheid biedt aan softwareproducenten om informatie in IFC weg te schrijven. Er kan dan gesteld worden of het niet beter is dat IAI in de toekomst deze vrijheid een beetje aan banden legt, zodat IFC in het algemeen vereenvoudigd kan worden.

Hoewel het IFC-bestand een bom aan informatie bevat, is er toch veel kennis van informatica nodig om deze informatie uit het bestand te halen en te verwerken. Het hoge niveau van programmeerkennis kan dus een struikelblok zijn voor toekomstige studenten die in dit thema willen verdergaan.

We kunnen concluderen dat de uitwerking van beide testcases in zeker opzicht geslaagd is. De eerste testcase genereert dezelfde tabellen als Revit, maar omdat dit in de code is ingebouwd. Hier kan de gebruiksvriendelijkheid van de opbouw van deze tabellen zeker verbeterd worden.

Zoals in het begin al aangegeven is dit een eerste aanzet naar het berekenen van een eenvoudige meetstaat. De tweede testcase is hier een voorbeeld van en kan zeker nog uitgebreid worden. Vooral de berekening van meer complexe geometrie moet zeker nog verder uitgewerkt worden. Want nu is het enkel mogelijk volumes te berekenen van rechte wanden en rechthoekige vloeren voorgesteld door extrusie. Tijdens het berekenen van de waardes werden ze vergeleken met de waardes uit Revit en de Solibri Model Viewer²⁴.

In het algemeen is de uitvoer van de gegevens naar Excel of PDF zeker geslaagd. De gegevens worden correct en op een eenvoudige en gebruiksvriendelijke manier weggeschreven, zodat deze gegevens verder kunnen gebruikt worden in het bouwproces.

²⁴ <http://www.solibri.com/solibri-model-viewer.html>

Bibliografie

1. **Berard, Ole en Schulz, Jesper.** *Geometrical Correction and Conversion of IFC2x3 Models.* Kopenhagen : IT University, 2008. p. 127, Proefschrift.
2. **Khemlani, Lachmi.** Top Criteria for BIM Solutions: AECbytes Survey Results. *AECbytes* . [Online] 10 Oktober 2007. [Citaat van: 12 Maart 2009.] <http://www.aecbytes.com/feature/2007/BIMSurveyReport.html>.
3. **BuildingSMART.** Learning Guide. *Information Delivery Manual.* [Online] 17 Januari 2009. http://www.iai.no/idm/idm_learning/idm_learning.htm.
4. **IFDWeb.** *IFDWeb.* [Online] 11 Mei 2009. <http://www.ifd-library.org/>.
5. **Böhms, Michel.** *BIM - Facts & Opinions — Actors & Results.* [Presentatie] 2008.
6. *ISO 10303-11:2004 Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language.*
7. *ISO 10303-28:2007 Industrial automation systems and integration -- Product data representation and exchange -- Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas.*
8. **ISG.** Certification. [Online] <http://www.iai.hm.edu/how-to-implement-ifc/certification>.
9. *IFC2x3 Final Documentation.* [Online] Model Support Group, 8 Juli 2007. <http://www.iai-tech.org/ifc/IFC2x3/TC1/html/index.htm>.
10. **de Laat, Ruben.** *BIMserver.* [interv.] Tim Lemoine. 13 Maart 2009.
11. **Bell, Håvard.** *Ifc 2x4 Goodies.* Washington DC : sn, 2008. p. 26.
12. **Kiviniemi, Arto.** *buildingSMART.* [interv.] Tim Lemoine. 8 Februari 2009.
13. **CPI.** *ik bim, jij bim-t, wij bim-men.* Delft : CPI, 2008. p. 33. ISBN/EAN 978-90-812963-1-1.
14. *Information Model.* [Online] 2009. [Citaat van: 14 April 2009.] <http://www.informationmodel.nl>.
15. **Cleempoel, Leslie Van.** *BIM: een nieuwe visie voor de bouwwereld.* Antwerpen : Artesis Hogeschool, 2008. p. 113, Proefschrift.
16. **Schrauwen, Kurt.** *Virtueel ontwerpen met IFC.* Antwerpen : Artesis Hogeschool, 2008. p. 83, Proefschrift.
17. **buildingSMART.** *buildingSMART.* [Online] BuildingSMART, 2008. <http://buildingsmart.com/>.

18. **Inc., CAD MicroSolutions.** The future of the Industry is BIM. *CAD MicroSolutions Inc.* [Online] 16 April 2009. <http://www.cadmicro.com/categories/c-pro-1-autodesk-b-BIM.htm>.
19. *IFC 2.x Edition 3.* [Online] STEP Tools, 2007.
http://www.steptools.com/support/stdev_docs/express/ifc2x3/html/.
20. **Campbell, Dace A.** *BIM - The Web Application for AEC.* [Presentatie] 2007.
21. *The buildingSMART Glossary of Terms.* [Document] sl : Model Support Group, 2007.
22. **Erabuild.** *Review of the Development and Implementation of IFC compatible BIM.* [Document] 2008.

