

中高層建築物におけるBIMを活用した木材利用の環境整備

木造BIMモデルパラメータガイド(木造設計BIMモデルにおける各オブジェクト データへの情報の持たせ方に関する説明書)

2025年3月

林野庁

0. 目次

- [0. 目次](#)
- [1. はじめに](#)
 - [1.1 本書について](#)
 - [1.2 用語の説明](#)
- [2. IFCサンプルデータについて](#)
 - [2.1 概要と使い方](#)
 - [2.1.1 情報の詳細度について](#)
 - [2.1.2 閲覧方法](#)
 - [2.2 構造モデル](#)
 - [2.3 意匠モデル](#)
- [3. IFCサンプルデータの作成方法](#)
 - [3.1 作成手順](#)
 - [3.2 ユースケースについて](#)
 - [3.3 パラメータ対照表について](#)
 - [3.3.1 IFCプロパティについて](#)
 - [3.4 パラメータ対照表\(構造モデル\)](#)
 - [3.5 パラメータ対照表\(意匠モデル\)](#)
 - [3.6 IFCパラメータマッピングテーブル](#)
- [4. IFCサンプルデータの活用方法](#)
 - [4.1 炭素貯蔵量の算定](#)
 - [4.1.1 出力フォーマット](#)
 - [4.1.2 出力結果](#)
 - [4.1.3 情報連携](#)
 - [4.2 材料調達連携](#)

- [4.2.1 テンプレートファイルの設定](#)
- [4.2.2 出力結果](#)
- [4.2.3 情報連携](#)
- [5. 参考プログラム](#)
 - [5.1 Pythonプログラムとテンプレートファイルの使用例](#)
 - [テンプレートファイルの記述方法](#)
 - [\[設定\] セクション](#)
 - [\[出力\] セクション](#)
 - [Pythonプログラムの例](#)

1. はじめに

1.1 本書について

本書は、木造建築物の設計におけるBIM(Building Information Modeling)活用の手法について、情報連携の観点から必要となる情報についてまとめ、かつ、実際のBIMモデルとしての情報の格納実例をまとめたものです。

また、BIMモデルの活用目的(ユースケース)のうち、「材料調達」と「炭素貯蔵量の算定」は、実際にIFCデータから情報を抽出し連携する例を掲載しています。なお、本書に記載する内容は、実際のプロジェクトに応じて適宜アレンジすることによりお役立てください。

- 本書では、木造建築物の設計におけるBIMモデルの活用目的として以下を挙げています。
 - 材料調達
 - 建築物に使用する木材・木質材料(以下「木材」という。)の寸法・強度等や概算量についての木材関連事業者への早期情報伝達
 - 省エネ計算(外皮・一次エネルギー)
 - 「非住宅建築物に関する省エネルギー基準に準拠したプログラム」への形状・属性情報伝達
 - 構造計算
 - 共通フォーマットを想定した構造計算プログラムへの形状・属性情報伝達
 - 炭素貯蔵量の算定
 - 炭素貯蔵量計算シートへの形状・属性情報伝達
 - (防耐火設計・避難安全検証)※
 - 防耐火仕様の納まり検討、監理への使用
 - 設計時点で検討される仕上げ材等の情報を基にした避難安全検証への活用

※ 防耐火設計については、現状ではBIMモデルの活用目的に応じた形状情報の詳細度の設定が特に重要と考えられることや、避難安全検証については、現状ではBIMモデルと直接データ連携が可能な市販の避難安全検証プログラムが存在しないこと等に鑑み、本書では、BIMモデルのパラメータに関する整理を行う観点から、防耐火設計・避難安全検証はBIMモデルの一般的な活用目的(ユースケース)の例として挙げるのみとしています。

なお、本書は、令和6年度林野庁委託事業「中高層建築物における木材利用の環境整備」において、学識経験者、業界団体、設計者、施工者、木材加工・流通関連事業者、ソフトウェア関連事業者等から成る委員会における検討を経て作成されました。

1.2 用語の説明

本書で用いる主な用語について説明します。

用語	説明
IFC (Industry Foundation Classes)	buildingSMART International が策定した建築・土木分野向けの国際標準データ形式。ソフトウェア間でのデータ連携を可能にする中立的なファイル形式
IFCエンティティ (IFC Entities)	IFCスキーマ内で定義される柱・梁・壁などのオブジェクトのクラス。例として IfcWall や IfcBeam などがあり、建築要素を統一的に扱うための枠組み
パラメータ (Parameter)	モデル要素に付随する個別の情報項目。本書では、モデル内の情報を特にパラメータと呼び、パラメータを含めた概念的な情報項目を情報と位置付けています
形状情報	柱や梁、壁などの要素の大きさ・位置・形状など、3次元モデルを構成するジオメトリに関する情報
属性情報	形状以外の情報。例えば材料やメーカー、仕上げ、積算や維持管理のための情報など、あらゆるテキスト・数値・区分などが含まれる
Property Set (プロパティセット)	IFC で定義される属性情報をグループ化したもの。例えば Pset_WallCommon のように、壁に共通して持たせるべき属性をまとめて管理するための枠組み
Property Name (プロパティ名)	Property Set 内に含まれる各属性の名称。例えば「Reference」や「FireRating」など
PredefinedType	IFCエンティティごとの「サブ分類」を示すプロパティ。例えば IfcColumn であれば、COLUMN や PILASTER のように用途に応じて細分化するために用いられる
ユースケース	BIMモデルを使って行う具体的な活用シナリオのこと
情報交換要件	建築に関わる情報の共有を行う上で、どのような情報をどのレベルで作成するかを事前に合意・定義すること。特にその作成にBIMモデルを用いる場合は、情報の入出力の基準や、IFCのエンティティやプロパティをどう扱うかなどを決める
データ連携	異なるソフトウェア同士がモデルや属性情報をやり取りすること。IFCなどの中立ファイル形式を用いることで、互換性を確保しやすい。本書では、情報連携のうち特にモデルデータの状態でやり取りを行うことをデータ連携としています
ネイティブデータ (Native Data)	各BIMソフトの専用形式で保存された元のデータ。IFC等に変換する前段階のファイルを指す

2. IFCサンプルデータについて

2.1 概要と使い方

- 本書では、BIMモデルとしてIFC (Industry Foundation Classes) の仕様で作成したデータファイルを想定しています。
- 本書で指すサンプルのIFCデータは、林野庁補助事業において作成された[中規模ビル3階建て事務所の木造化標準モデル\(令和4年度版\)](#)に基づき作成したものです。
 - ただし、前述した標準モデル資料の時点では記載されていないあるいは未決定の情報については、本書の作成用に適宜独自に付け足しており、情報の持ち方の参考にはいただけますが、実際の設計図書程度の精度ではないことに注意してください。
- IFCデータを中心にした情報連携並びにデータ連携を行う場合、IFCデータの作成側とその受け取り側で、情報をどのように入力・作成するかという定義 (=情報交換要件) を予め行うことが重要です。
本書およびサンプルIFCデータの使い方として、その情報交換要件を定める上での参考事例として参照・活用されることを想定しています。

2.1.1 情報の詳細度について

- 形状情報については、ユースケースに要する必要最小限の詳細度としています。
- 属性情報については、ユースケースに要する必要最小限の詳細度としています。

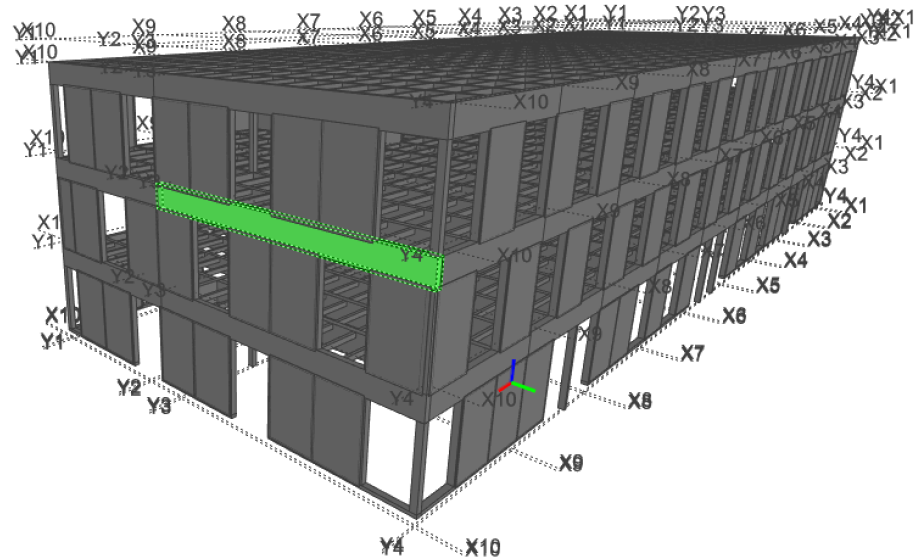
2.1.2 閲覧方法

- サンプルのIFCデータは、.ifcの拡張子のファイルをIFCビューワーで開くことで属性情報および形状情報を見ることが出来ます。
 - IFCビューワーの例:
 - BIMVision
 - FZKViewer
 - Bonsai (Blenderのアドオンソフト。旧BlenderBIM)
- サンプルのIFCデータには、構造モデルと意匠モデルの二種類があり、[3.4 パラメータ対照表](#)において各ユースケースに対応する情報を格納した場合の参考例を示しています。
出来る限り全てのプロパティについて値を入力していますが、これらは当然全て入力される必要はなく、ユースケースや関係者間の合意に合わせて、「欲しい情報」・「入力できる情報」という観点から適宜取捨選択されたうえで、BIMモデルとして作成されることを前提としています。

2.2 構造モデル

柱:IfcColumn、梁:IfcBeam、耐力壁:IfcWall、構造床:IfcSlab(面材による剛床範囲)

をオブジェクトとして配置しており、対応する情報を格納しています。ここでは、IFCビューワーのBIMVisionで開いたときのキャプチャを示しています。

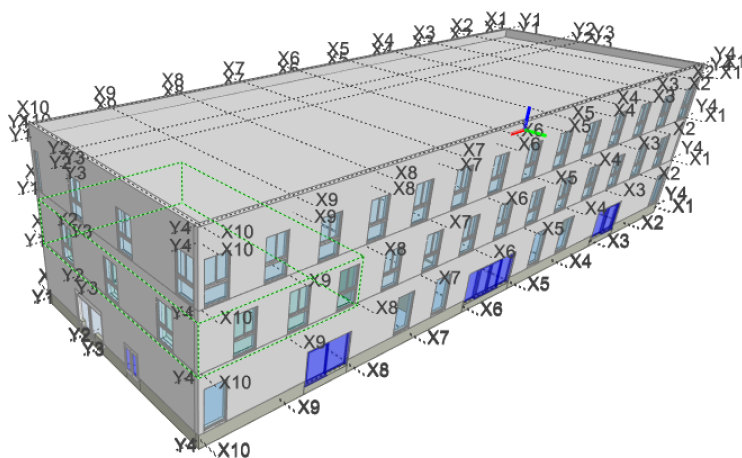


Properties	Location	Classification	Relations	
	Name	Value		
	Element Specific			
	Guid	30Yc9U1ML1DRn4bzS1kHGT		
	IfcEntity	IfcBeam		
	Name	木梁:210x840:162302		
	ObjectType	木梁:210x840		
	PredefinedType	BEAM		
	Tag	162302		
	Profile			
	ProfileName	210x840		
	XDim	840		
	YDim	210		
	JPPset_TimberElementGeneral			
	Adhesives	レゾルシノール樹脂及びレゾルシノール・フェノール樹脂(積層・たて継ぎ)		
	ApprovedQuality	C-2 防腐・防蟻処理構造用集成材-2		
	EnvironmentalCondition	使用環境A		
	FormaldehydeEmissions	F☆☆☆☆		
	PreservationAgent	ACC-1		
	PreservingMethod	K3		
	ProductName	JAS構造用集成材		
	StrengthClass	E105-F300(異等級構成)		
	SurfaceQuality	2種		
	TimberSpecies	カラマツ		
	JPPset_TimberFrameDimension			
	Length	10 920		
	StartingStorey	3FL		
	Pset_BeamCommon			
	IsExternal	No		
	LoadBearing	Yes		
	Reference	210x840		
	Roll	0		
	Slope	0		
	Span	10 920		
	Pset_BeamCommon			
	IsExternal	No		
	LoadBearing	Yes		
	Pset_ConcreteElementGeneral			
	StrengthClass	E105-F300		
	Pset_EnvironmentalImpactIndicators			

2.3 意匠モデル

壁:IfcWall、窓:IfcWindow、ドア:IfcDoor、床:IfcSlab、屋根:IfcSlab、部屋:IfcSpace(空間)

をオブジェクトとして配置しており、対応する情報を格納しています。ここでは、IFCビューワーのBIMVisionで開いたときのキャプチャを示しています。



Properties	Location	Classification	Relations
Name	Value	Unit	
Element Specific			
CompositionType	ELEMENT		
Guid	39ngMQhz07wMV8tZg1083		
IfcEntity	IfcSpace		
LongName	事務室		
Name	96		
PredefinedType	NOTDEFINED		
Profile			
Pset_SpaceCommon			
Storey	2FL		
Pset_SpaceCommon			
IsExternal	No		
Reference	事務室 96		
Pset_SpaceCoveringRequirements			
CeilingCovering	ロックウール化粧吸音板		
FloorCovering	タイルカーペットB		
WallCovering	石膏ボード t12.5		
Pset_SpaceHeaterTypeCommon			
Reference	事務室 96		
Pset_SpaceHVACDesign			
AirConditioning	Yes		
AirHandlingName	PAC-2-1		
ArtificialLighting	Yes		
HotWaterPresence	Yes		
MechanicalVentilation	No		
Pset_SpaceLightingDesign			
LightingFixtures	FRS18L3V-322		
NumberOfLightingFixtures	42		
Pset_SpaceOccupancyRequirements			
OccupancyType	事務室		
Qto_SpaceBaseQuantities			
FinishCeilingHeight	816.864	mm	
Height	2680	mm	
NetPlannedArea	188.23623	m2	

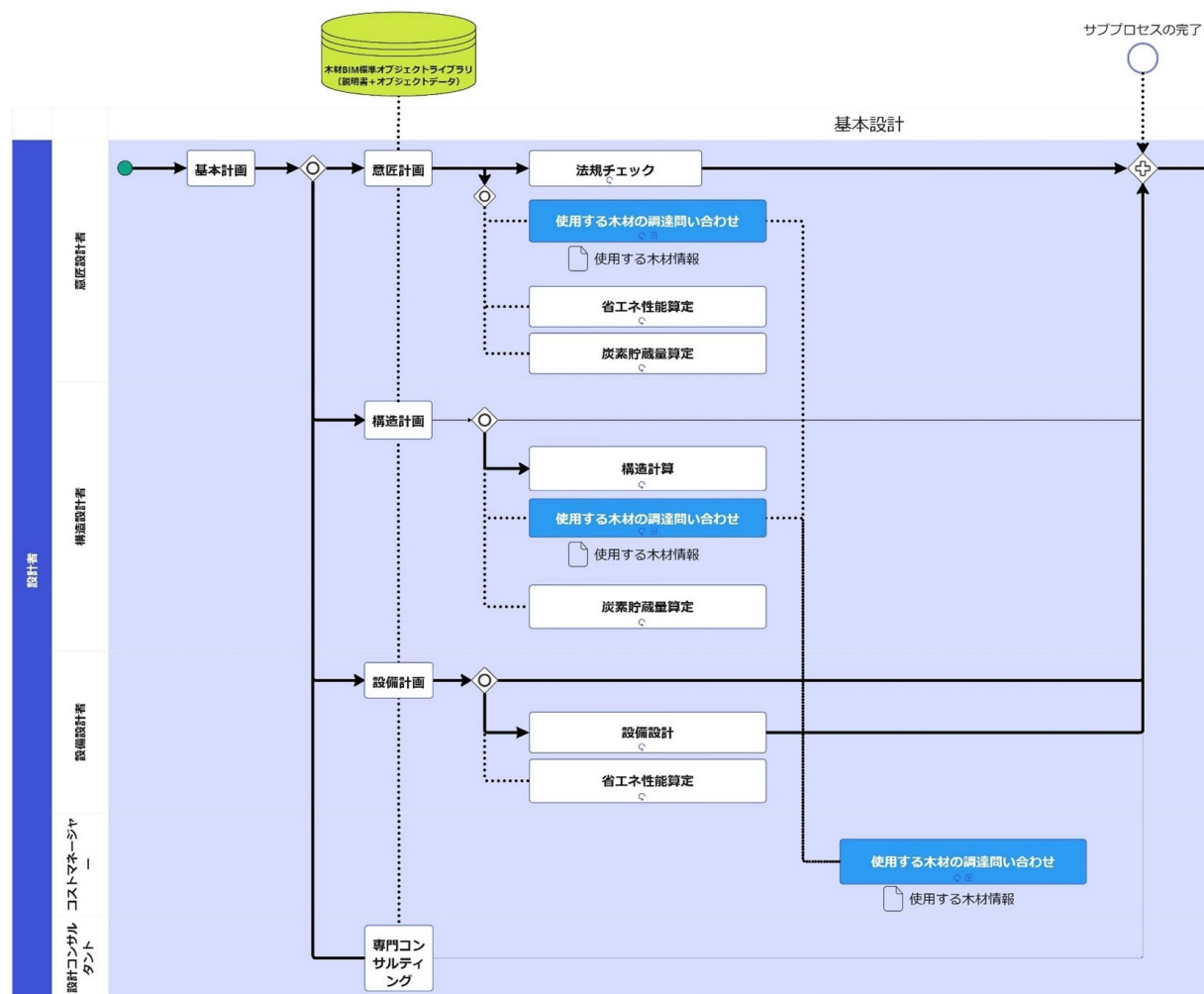
両方とも、IfcWall等のエンティティ(オブジェクト)に対して、その属性情報として情報を持つか、プロパティセットとして属性情報を与えています。また、使用する材料の詳細については、そのオブジェクトに関連する情報として別に定義付けています。

3. IFCサンプルデータの作成方法

3.1 作成手順

本書で挙げているユースケースは設計～申請前段階での活用を想定しているため、原則、基本設計や実施設計の中で適宜設計検討の為に行為れることが理想的です。

別資料の「木材調達等のための情報伝達の標準プロセス」の「2.3 プロセスマップ」において示すように、一般的には意匠計画や構造計画の時点でBIMソフトによるネイティブデータを作成されるものと考えられます。



上記の図の「意匠計画」や「構造計画」の時点で、下記の図のような手順で作成されるものと想定しています。



1. BIMソフトに形状・属性を入力

- 本書の[3.4 パラメータ対照表](#)を参考に、ユースケースに必要な情報を入力。必ずしも同じ名称でパラメータを設ける必要はなく、管理のしやすい命名規則や、既に使用されているパラメータに重複するものがあれば、それを用いても問題ありません。
- 木材調達のユースケースにあたっては、入力する情報の参考として「木材調達等のための情報伝達の標準プロセス」の「3 情報交換要件」に参考例を示しています。

2. BIMソフトからIFCデータへ書き出す

- 本書の[3.4 パラメータ対照表](#)を参考に、指定するIFCエンティティ、プロパティに出力します。
- Revitであれば、[3.5 IFCパラメータマッピングテーブル](#)を読み込むことで[3.4 パラメータ対照表](#)に対応した出力が可能です。

3. 目的に応じて情報の抽出などを行う

- IFCデータから別の書式に変換する場合、情報の抽出やファイルフォーマットの変換が必要です。
- 本書では[IFCサンプルデータの活用方法](#)において、[4.1 炭素貯蔵量の算定](#)と、[4.2 材料調達連携](#)のユースケースの為の情報抽出方法について実践一例を掲載しています。

3.2 ユースケースについて

- ユースケース(IFCデータの活用目的)によって、必要な情報は変わります。
[3.4 パラメータ対照表\(構造モデル\)](#)および[3.5 パラメータ対照表\(意匠モデル\)](#)における表記は以下の通りとします。「○」が付いている情報が、その用途で連携対象になる情報を示しています。
 - 材料調達: 材
 - 省エネ計算: エ
 - 構造計算: 構
 - 炭素貯蔵量の算定: C

3.3 パラメータ対照表について

- [3.4 パラメータ対照表](#)では、一般的な部位名称(=部位)と、一般的な情報名称(=情報)に対し、BIMソフト(Revitを想定)でのパラメータ名称と分類(=Revit大分類,Revit中分類,Revit小分類)、さらにIFCサンプルデータで対応するIFCのオブジェクト名(=IFC Entities)と、定義済タイプ(=PredefinedType)、プロパティセット(=Property Set,Property Name)を表に整理しています。
- なお、パラメータの入力の手数を減らす観点から、Revitでのパラメータについて下記の方針としています。
 - 組み込み (Built-In) パラメータを使用
 - 設計三会及びRUG(Revit User Group)で提供されるテンプレート等のパラメータを使用
 - 必要に応じて新規のパラメータを追加
- また、特記の無いパラメータについては、今回想定したユースケースには直接関与しないものとし、省略しています。

3.3.1 IFCプロパティについて

- 各パラメータを「IFC4X3 ADD2」のスキーマのプロパティセットに対照させる形で表にまとめています。
 - 既存のプロパティセットに対応するものが無いと考えられるパラメータについては、プロパティセットの拡張例として独自に赤字のプロパティを追加していますが、現在、bSI(buildingSMART International)において正式に対応しているものではありません。
 - ただ、本事業の主題である木材利用促進の観点から、特にJPPset_の接頭で始まるプロパティセットについては、木材に関する共通のIFCプロパティセットとして、本事業における成果の一部として積極的に提案するものです。(木材調達、炭素貯蔵量の算定、構造連携のユースケースにおいて特に有用であると考えられます)
 - 表中の「Property Set」では、特に指定の無い限りプロパティセットが IfcRelDefinesByProperties によって要素に付与されるものとしています。
 - ただし、別途IfcRelContainedInSpatialStructure (空間階層への包含)や、IfcRelAssociatesMaterial (材料情報との関連付け) 等が記述されている場合は、リレーションシップによる情報の紐づけであると読み替えてください。
- ここで提案するJPPset_プロパティセットについては、今後変更・修正の可能性があります。

3.4 パラメータ対照表(構造モデル)

ここでは、材料調達、構造計算、炭素貯蔵量の算定に用いられる属性情報等を構造モデルのIFCデータに格納した場合の参考例を示しています。また、本書において木材のIFCプロパティセットとして提案するものを「JPPset_」として以下に示しています。

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
柱	部位	構造 柱	その他	定義済 み IFC ク ラス	柱,間柱,地束,床 束,小屋 束:IfcColumn たて 枠:IfcMember	IfcColumn IfcMember	-	-	-	-	-	-	○	-
柱	部位	構造 柱	その他	定義済 み IFC タ イプ	柱,間柱:COLUMN 地束,床束,小屋 束:STANDCOLUMN たて枠:STUD	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	-	-	-	-	-	○	-
柱	幅	構造 柱	構造 断面 ジオメ トリ	幅	120	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberFrameDimension※2	Width	Length	○	-	○	○
柱	せい	構造 柱	構造 断面 ジオメ トリ	高さ	120	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberFrameDimension	Height	Length	○	-	○	○
柱	(位置 座標)	構造 柱	拘束	基準レベ ル	1FL	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberFrameDimension	StartingStorey (IfcBuildingStorey.Nameと同値)	Label	-	-	○	-
柱	(位置 座標)	構造 柱	拘束	基準レベ ル オフ セット	420	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	-	-	-	-	-	○	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
柱	(位置座標)	構造柱	拘束	上部レベル	2FL	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberFrameDimension	EndStorey (IfcBuildingStorey.Nameと同値)	Label	-	-	○	-
柱	(位置座標)	構造柱	拘束	上部レベル オフ セット	-230	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	-	-	-	-	-	○	-
柱	長さ	構造柱	寸法	長さ	3230.0	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	Qto_ColumnBaseQuantities	Length	Length	○	-	-	○
柱	数量	構造柱	寸法	容積	0.09197 m³	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	Qto_ColumnBaseQuantities	NetVolume	Volume	○	-	-	○
柱	調達寸法※1	構造柱	その他	調達寸法	120x120x3650	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral※3	OriginalLength※5	Label	○	-	-	○
柱	品名※1	構造柱	その他	品名	JAS構造用集成材	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	ProductName	Label	○	-	○	○
柱	樹種※1	構造柱	その他	樹種	カラマツ	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	TimberSpecies	Label	○	-	○	○
柱	強度等級※1	構造柱	その他	強度等級	E95-F315 (同一等級)	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	StrengthClass	Label	○	-	○	-

部 位	情 報	Revit 大分 類	Revit 中分 類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
柱	接 着 性 能 ※1	構造 柱	その他	接着性能	使用環境A	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	EnvironmentalCondition	Label	○	-	-	-
柱	使用 接 着 剤 ※1	構造 柱	その他	使用接着剤	レゾルシノール樹脂 及びレゾルシノール・ フェノール樹脂（積 層・たて継ぎ）	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	Adhesives	Label	○	-	-	-
柱	材 面 の 品 質 ※1	構造 柱	その他	材面の品質	2種	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	SurfaceQuality	Label	○	-	-	-
柱	ホル ム アル デ ヒド 放 散 量 ※1	構造 柱	その他	ホルムアルデヒド 放散量	F☆☆☆☆	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	FormaldehydeEmissions	Label	○	-	-	-
柱	乾 燥 処 理 ※1	構造 柱	その他	乾燥処理		IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	DryingProcess	Label	○	-	-	-

部 位	情 報	Revit 大分 類	Revit 中分 類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
柱	含 水 率 ※1	構造 柱	その他	含水率		IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	MoistureContent	Label	○	-	-	-
柱	保 存 処 理 ※1	構造 柱	その他	保存処理	K3	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	PreservingMethod	Label	○	-	-	-
柱	木 材 保 存 剤 ※1	構造 柱	その他	木材保存剤	ACC-1	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	PreservationAgent	Label	○	-	-	-
柱	AQ 認 証 区 分 ※1	構造 柱	その他	AQ認証区分	C-2	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	ApprovedQuality	Label	○	-	-	-
柱	製 造 者 名 ※1	構造 柱	その他	製造者名	***	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberElementGeneral	Manufacturer (Pset_ManufacturerTypeInformation.Manufacturer としてもよい)	Label	○	-	-	-
柱	始 端 接 合	構造 柱	その他	StartJoint	PB-33	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberConnection※4	ConnectionDetailsStarting	Label	-	-	○	-
柱	終 端 接 合	構造 柱	その他	EndJoint	PB-33	IfcColumn IfcMember	COLUMN STANDCOLUMN STUD	JPPset_TimberConnection	ConnectionDetailsEnd	Label	-	-	○	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
横 架 材	部 位	構 造 フ レ ーム	そ の 他	定 義 済 み IFC ク ラス	梁,桁,土台,大 引:IfcBeam 母屋,棟木,垂木,隅 木,筋違,火 打:IfcMember	IfcBeam IfcMember	-	-	-	-	-	-	○	-
横 架 材	部 位	構 造 フ レ ーム	そ の 他	定 義 済 み IFC タ イ プ	梁,桁:BEAM 土台,大引:JOIST 母屋,棟 木:PURLIN 垂木,隅 木:RAFTER 筋違,火打:BRACE	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	-	-	-	-	-	○	-
横 架 材	幅	構 造 フ レ ーム	構 造 断 面 ジオメ トリ	幅	240	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberFrameDimension	Width	Length	○	-	○	○
横 架 材	せ い	構 造 フ レ ーム	構 造 断 面 ジオメ トリ	高さ	840	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberFrameDimension	Height	Length	○	-	○	○
横 架 材	(位 置 座 標)	構 造 フ レ ーム	拘 束	参 照 レ ベ ル	2FL	IfcBeam	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberFrameDimension (IfcRelContainedInSpatialStructure でリレーションを参照してもよい)	StartingStorey (IfcBuildingStorey.Nameと同値) (IfcBuildingStorey.Nameとしてもよい)	Label	-	-	○	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
横 架 材	(位 置 座 標)	構造 フレ ーム	拘束	始端レベ ル オフ セット	-230	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	-	-	-	-	-	○	-
横 架 材	(位 置 座 標)	構造 フレ ーム	拘束	終端レベ ル オフ セット	-230	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	-	-	-	-	-	○	-
横 架 材	長 さ	構造 フレ ーム	寸法	長さ	5460.0	IfcBeam	BEAM JOIST PURLIN RAFTER BRACE	Pset_BeamCommon	Span	Length	○	-	-	○
横 架 材	数 量	構造 フレ ーム	寸法	容積	1.101 m ³	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	Qto_BeamBaseQuantities	NetVolume	Volume	○	-	-	○
横 架 材	調 達 寸 法 ※1	構造 フレ ーム	そ 他	調達寸 法	240x840x7000	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	OriginalLength※6	Label	○	-	-	○
横 架 材	品 名 ※1	構造 フレ ーム	そ 他	品名	JAS構造用集成材	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	ProductName	Label	○	-	○	○

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
横 架 材	樹 種 ※1	構造 フレ ーム	その 他	樹種	カラマツ	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	TimberSpecies	Label	○	-	○	○
横 架 材	強 度 等 級 ※1	構造 フレ ーム	その 他	強度等 級	E105-F300 (異 等級構成)	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	StrengthClass	Label	○	-	○	-
横 架 材	接 着 性 能 ※1	構造 フレ ーム	その 他	接着性 能	使用環境A	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	EnvironmentalCondition	Label	○	-	-	-
横 架 材	使 用 接 着 剤 ※1	構造 フレ ーム	その 他	使用接 着剤	レゾルシノール樹脂 及びレゾルシノール・ フェノール樹脂（積 層・たて継ぎ）	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	Adhesives	Label	○	-	-	-
横 架 材	材 面 の 品 質 ※1	構造 フレ ーム	その 他	材面の品 質	2種	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	SurfaceQuality	Label	○	-	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
横 架 材	ホルム アル デヒド 放散量 ※1	構造 フレーム	その他	ホルムアル デヒド 放散量	F☆☆☆☆	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	FormaldehydeEmissions	Label	○	-	-	-
横 架 材	乾燥 処理 ※1	構造 フレーム	その他	乾燥処理		IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	DryingProcess	Label	○	-	-	-
横 架 材	含水率 ※1	構造 フレーム	その他	含水率		IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	MoistureContent	Label	○	-	-	-
横 架 材	保存 処理 ※1	構造 フレーム	その他	保存処理	K3	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	PreservingMethod	Label	○	-	-	-
横 架 材	木材 保存 剤 ※1	構造 フレーム	その他	木材保存 剤	ACC-1	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	PreservationAgent	Label	○	-	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
横 架 材	AQ 認 証 区 分 ※1	構造 フレ ーム	その他	AQ認証 区分	C-2	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	ApprovedQuality	Label	○	-	-	-
横 架 材	製 造 者 名 ※1	構造 フレ ーム	その他	製造者 名	***	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberElementGeneral	Manufacturer (Pset_ManufacturerTypeInformation.Manufacturer としてもよい)	Label	○	-	-	-
横 架 材	始 端 接 合	構造 フレ ーム	その他	StartJoint	BH-255	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberConnection	ConnectionDetailsStarting	Label	-	-	○	-
横 架 材	終 端 接 合	構造 フレ ーム	その他	EndJoint	BH-255	IfcBeam IfcMember	BEAM JOIST PURLIN RAFTER BRACE	JPPset_TimberConnection	ConnectionDetailsEnd	Label	-	-	○	-
面 材 耐 力 壁	部 位	壁	その他	定義済 み IFC ク ラス	IfcWall	IfcWall	-	-	-	-	-	-	○	-
面 材 耐 力 壁	部 位	壁	その他	定義済 み IFC タ イプ	耐力壁:SHEAR(水 平力への抵抗要 素)	IfcWall	SHEAR	-	-	-	-	-	○	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
面材耐力壁	幅	壁	構成	幅	18	IfcWall	SHEAR	Qto_WallBaseQuantities	Width	Length	○	-	○	○
面材耐力壁	(位置座標)	壁	拘束	基準レベル	1FL	IfcWall	SHEAR	IfcRelContainedInSpatialStructure でリレーションを参照	IfcBuildingStorey.Name	Label	-	-	○	-
面材耐力壁	長さ	壁	寸法	長さ	1820mm	IfcWall	SHEAR	Qto_WallBaseQuantities	Length	Length	○	-	-	○
面材耐力壁	数量	壁	寸法	容積	0.105 m ³	IfcWall	SHEAR	Qto_WallBaseQuantities	NetVolume	Volume	○	-	-	○
面材耐力壁	調達寸法※1	壁	その他	調達寸法	910×1820×18mm	IfcWall	SHEAR	JPPset_TimberElementGeneral	OriginalLength	Label	○	-	-	○
面材耐力壁	品名※1	壁	その他	品名	JAS構造用合板	IfcWall	SHEAR	JPPset_TimberElementGeneral	ProductName	Label	○	-	○	○

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
面材耐力壁	樹種※1	壁	その他	樹種	カラマツ	IfcWall	SHEAR	JPPset_TimberElementGeneral	TimberSpecies	Label	○	-	○	○
面材耐力壁	等級※1	壁	その他	等級	2級	IfcWall	SHEAR	JPPset_TimberElementGeneral	StrengthClass	Label	○	-	○	-
面材耐力壁	接着の性能※1	壁	その他	接着の性能	特類	IfcWall	SHEAR	JPPset_TimberElementGeneral	EnvironmentalCondition	Label	○	-	-	-
面材耐力壁	板面の品質※1	壁	その他	板面の品質	2種	IfcWall	SHEAR-	JPPset_TimberElementGeneral	SurfaceQuality	Label	○	-	-	-
面材耐力壁	ホルムアルデヒド放散量※1	壁	その他	ホルムアルデヒド放散量	F☆☆☆☆	IfcWall	SHEAR	JPPset_TimberElementGeneral	FormaldehydeEmissions	Label	○	-	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
面材耐力壁	曲げ性能※1	壁	その他	曲げ性能		IfcWall	SHEAR	JPPset_TimberElementGeneral	FlexuralStrength	Label	○	-	-	-
面材耐力壁	防虫剤※1	壁	その他	防虫剤		IfcWall	SHEAR	JPPset_TimberElementGeneral	InsectRepellent	Label	○	-	-	-
面材耐力壁	保存処理※1	壁	その他	保存処理		IfcWall	SHEAR	JPPset_TimberElementGeneral	PreservingMethod	Label	○	-	-	-
面材耐力壁	木材保存剤※1	壁	その他	木材保存剤		IfcWall	SHEAR	JPPset_TimberElementGeneral	PreservationAgent	Label	○	-	-	-
面材耐力壁	AQ認証区分※1	壁	その他	AQ認証区分		IfcWall	SHEAR	JPPset_TimberElementGeneral	ApprovedQuality	Label	○	-	-	-
面材耐力壁	製造者名※1	壁	その他	製造者名		IfcWall	SHEAR	JPPset_TimberElementGeneral	Manufacturer (Pset_ManufacturerTypeInfo.Manufacturerとしてもよい)	Label	○	-	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
構造床	部位	床	その他	定義済み IFC クラス	IfcSlab	IfcSlab	-	-	-	-	-	-	○	-
構造床	部位	床	その他	定義済み IFC タイプ	水平面:FLOOR 屋根面:ROOF	IfcSlab	FLOOR ROOF	-	-	-	-	-	○	-
構造床	幅	床	構成	幅	18	IfcSlab	FLOOR ROOF	Qto_SlabBaseQuantities	Width	Length	○	-	○	○
構造床	(位置座標)	床	拘束	基準レベル	1FL	IfcSlab	FLOOR ROOF	IfcRelContainedInSpatialStructure でリレーションを参照	IfcBuildingStorey.Name	Label	-	-	○	-
構造床	長さ	床	寸法	長さ	1820mm	IfcSlab	FLOOR ROOF	Qto_IfcSlabBaseQuantities	Length	Length	○	-	-	○
構造床	数量	床	寸法	容積	0.105 m ³	IfcSlab	FLOOR ROOF	Qto_IfcSlabBaseQuantities	NetVolume	Volume	○	-	-	○
構造床	調達寸法※1	床	その他	調達寸法	910×1820×18mm	IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	OriginalLength	Label	○	-	-	○
構造床	品名※1	床	その他	品名	JAS構造用合板	IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	ProductName	Label	○	-	○	○
構造床	樹種※1	床	その他	樹種	カラマツ	IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	TimberSpecies	Label	○	-	○	○

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
構造床	等級※1	床	その他	等級	2級	IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	StrengthClass	Label	○	-	○	-
構造床	接着の性能※1	床	その他	接着の性能	特類	IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	EnvironmentalCondition	Label	○	-	-	-
構造床	板面の品質※1	床	その他	板面の品質	2種	IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	SurfaceQuality	Label	○	-	-	-
構造床	ホルムアルデヒド放散量※1	床	その他	ホルムアルデヒド放散量	F☆☆☆☆	IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	FormaldehydeEmissions	Label	○	-	-	-
構造床	曲げ性能※1	床	その他	曲げ性能		IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	FlexuralStrength	Label	○	-	-	-
構造床	防虫剤※1	床	その他	防虫剤		IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	InsectRepellent	Label	○	-	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	工	構	C
構造床	保存処理 ※1	床	その他	保存処理		IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	PreservingMethod	Label	○	-	-	-
構造床	木材保存剤 ※1	床	その他	木材保存剤		IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	PreservationAgent	Label	○	-	-	-
構造床	AQ 認証区分 ※1	床	その他	AQ認証区分		IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	ApprovedQuality	Label	○	-	-	-
構造床	製造者名 ※1	床	その他	製造者名		IfcSlab	FLOOR ROOF	JPPset_TimberElementGeneral	Manufacturer (Pset_ManufacturerTypeInfo.Manufacturer としてもよい)	Label	○	-	-	-

※1:JAS表示項目

※2:JPPset_TimberFrameDimension:木材関連の情報連携の為のプロパティセットとして本事業で提案するもの。主に幅とせいや位置の幾何形状情報を定義する。

※3:JPPset_TimberElementGeneral:木材関連の情報連携の為のプロパティセットとして本事業で提案するもの。主に木材の属性情報を定義する。

※4:JPPset_TimberConnection:木材関連の情報連携の為のプロパティセットとして本事業で提案するもの。主に木材の端部接合情報を定義する。

※5:木取を示す語句の英訳が無いため、ここでは元の製材の長さを意図し「OriginalLength」というプロパティを追加

3.5 パラメータ対照表(意匠モデル)

ここでは、省エネ計算に用いられる属性情報等を、意匠モデルのIFCデータに格納した場合の参考例を示しています。

また、前述した「JPPset_」のプロパティセットと同様に、必要に応じて既存のプロパティセット等を拡張していますが、あくまで参考例であり、本書ではその拡張方法までを提案するものではないことに注意してください。

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
壁	(位置座標)	壁	その他	定義済み IFC クラス	IfcWall	IfcWall	-	-	-	-	-	○	-	-
壁	(位置座標)	壁	その他	定義済み IFC タイプ			-	-	-	-	-	○	-	-
壁	全厚	壁	構成	幅	267mm	IfcWall	-	Qto_WallBaseQuantities	Width	Length	-	○	-	-
壁	建材名称	壁	構成	構造[*]マテ リアル	せっこうボード	IfcWall	-	IfcRelAssociatesMaterialでリレーシ ョンを参照	IfcMaterialLayer.Name	Label	-	○	-	-
壁	建材番号	壁	構成	構造[*]マテ リアル>建材 番号 (マテリアルプ ロパティ)	61	IfcWall	-	IfcRelAssociatesMaterialでリレーシ ョンを参照	IfcMaterialLayer.Description	Label	-	○	-	-
壁	建材厚	壁	構成	構造[*]厚み	15mm	IfcWall	-	IfcRelAssociatesMaterialでリレーシ ョンを参照	IfcMaterialLayer.LayerThickness	Length	-	○	-	-
壁	壁の種類	壁	その他	外壁 接地壁	Yes No	IfcWall	-	Pset_WallCommon	isExternal (YesをTrueとして扱う)	Boolean	-	○	-	-
壁	壁の名称	壁	その他	外壁名称	OW1	IfcWall	-	-	Name	Label	-	○	-	-
壁	熱貫流率	壁	解析用プ ロパティ	熱伝達係数 (U)	1.00 直接、熱貫流率を 指定する場合	IfcWall	-	Pset_WallCommon	ThermalTransmittance	Double	-	○	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
壁	外皮面積 (窓含)	壁	寸法	面積	94.5	IfcWall	-	Qto_WallBaseQuantities	NetSideArea	Area	-	○	-	-
屋根	(位置座標)	屋根	その他	定義済み IFC クラス	IfcSlab	IfcSlab※9	-	-	-	-	-	○	-	-
屋根	(位置座標)	屋根	その他	定義済み IFC タイプ	IfcSlabとした場合 ROOF		-	-	-	-	-	○	-	-
屋根	全厚	屋根	構成	既定の厚さ	100mm	IfcSlab	ROOF	Qto_SlabBaseQuantities	Depth	Length	-	○	-	-
屋根	建材名称	屋根	構成	構造[*]マテ リアル	硬質ウレタンフォーム	IfcSlab	ROOF	IfcRelAssociatesMaterialでリレーシ ョンを参照	IfcMaterialLayer.Name	Label	-	○	-	-
屋根	建材番号	屋根	構成	構造[*]マテ リアル>建材 番号 (マテリアルプ ロパティ)	201	IfcSlab	ROOF	IfcRelAssociatesMaterialでリレーシ ョンを参照	IfcMaterialLayer.Description	Label	-	○	-	-
屋根	建材厚	屋根	構成	構造[*]厚み	70mm	IfcSlab	ROOF	IfcRelAssociatesMaterialでリレーシ ョンを参照	IfcMaterialLayer.LayerThickness	Length	-	○	-	-
屋根	屋根の名称	屋根	その他	屋根名称	(屋根) 陸屋根	IfcSlab	ROOF	-	Name	Label	-	○	-	-
屋根	熱貫流率	屋根	解析用プ ロパティ	熱伝達係数 (U)	1.00 直接、熱貫流率を 指定する場合	IfcSlab	ROOF	Pset_SlabCommon	ThermalTransmittance	Double	-	○	-	-
屋根	外皮面積	屋根	寸法	面積	803.56	IfcSlab	ROOF	Qto_SlabBaseQuantities	NetArea	Area	-	○	-	-
床	(位置座標)	床	その他	定義済み IFC クラス	IfcSlab	IfcSlab	-	-	-	-	-	○	-	-
床	(位置座標)	床	その他	定義済み IFC タイプ			-	-	-	-	-	○	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
床	全厚	床	構成	既定の厚さ	100mm	IfcSlab	-	Qto_SlabBaseQuantities	Depth	Length	-	○	-	-
床	建材名称	床	構成	構造[*]マテリアル	硬質ウレタンフォーム	IfcSlab	-	IfcRelAssociatesMaterialでリレーションを参照	IfcMaterialLayer.Name	Label	-	○	-	-
床	建材番号	床	構成	構造[*]マテリアル>建材番号 (マテリアルプロパティ)	201	IfcSlab	-	IfcRelAssociatesMaterialでリレーションを参照	IfcMaterialLayer.Description	Label	-	○	-	-
床	建材厚	床	構成	構造[*]厚み	70mm	IfcSlab	-	IfcRelAssociatesMaterialでリレーションを参照	IfcMaterialLayer.LayerThickness	Length	-	○	-	-
床	床の名称	床	その他	床名称	一般床	IfcSlab	-	-	Name	Label	-	○	-	-
床	熱貫流率	床	解析用プロパティ	熱伝達係数(U)	1.00 直接、熱貫流率を指定する場合	IfcSlab	-	Pset_SlabCommon	ThermalTransmittance	Double	-	○	-	-
床	外皮面積	床	寸法	面積	803.56	IfcSlab	-	Qto_SlabBaseQuantities	NetArea	Area	-	○	-	-
窓	(位置座標)	窓	その他	定義済み IFC クラス	IfcWindow	IfcWindow	-	-	-	-	-	○	-	-
窓	(位置座標)	窓	その他	定義済み IFC タイプ			-	-	-	-	-	○	-	-
窓	開口部名称	窓	タイプ	-	W550×H1850	IfcWindow	-	-	Name	Label	-	○	-	-
窓	熱貫流率	窓	解析用プロパティ	熱伝達係数(U)	1.00	IfcWindow	-	Pset_WindowCommon	ThermalTransmittance	Double	-	○	-	-
窓	日射熱取得率	窓	解析用プロパティ	日射熱取得率	1.00	IfcWindow	-	Pset_DoorWindowGlazingType	SolarHeatGainTransmittance	Double	-	○	-	-
窓	建具の種類	窓	拘束	(建具)建具種別	金属木複合製(単板ガラス)	IfcWindow	-	IfcRelDefinesByTypeでリレーションを参照	IfcWindowType.Name	Label	-	○	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
窓	ガラスの種類	窓	マテリアルと仕上げ	ガラスマテリアル	T	IfcWindow	-	IfcRelAssociatesMaterialでリレーションを参照	IfcMaterialLayer.Description	Label	-	○	-	-
窓	建具主寸法W	窓	寸法	幅	550mm	IfcWindow	-	-	OverallWidth	Length	-	○	-	-
窓	建具主寸法H	窓	寸法	高さ	1850mm	IfcWindow	-	-	OverallHeight	Length	-	○	-	-
窓	ブラインドの有無	窓	その他	ブラインドの有無	Yes	IfcWindow	-	Pset_WindowCommon	HasLouver※1	Boolean	-	○	-	-
ドア	(位置座標)	ドア	その他	定義済み IFC クラス	IfcDoor	IfcDoor	-	-	-	-	-	○	-	-
ドア	(位置座標)	ドア	その他	定義済み IFC タイプ			-	-	-	-	-	○	-	-
ドア	開口部名称	ドア	タイプ	-	W550xH1850	IfcDoor	-	-	Name	Label	-	○	-	-
ドア	熱貫流率	ドア	解析用プロパティ	熱伝達係数 (U)	1.00	IfcDoor	-	Pset_DoorCommon	ThermalTransmittance	Double	-	○	-	-
ドア	日射熱取得率	ドア	解析用プロパティ	日射熱取得率	1.00	IfcDoor	-	Pset_DoorWindowGlazingType	SolarHeatGainTransmittance	Double	-	○	-	-
ドア	建具の種類	ドア	拘束	(建具)建具種別	金属木複合製 (単板ガラス)	IfcDoor	-	IfcRelDefinesByTypeでリレーションを参照	IfcDoorType.Name	Label	-	○	-	-
ドア	ガラスの種類	ドア	マテリアルと仕上げ	ガラスマテリアル	T	IfcDoor	-	IfcRelAssociatesMaterialでリレーションを参照	IfcMaterialLayer.Description	Label	-	○	-	-
ドア	建具主寸法W	ドア	寸法	幅	550mm	IfcDoor	-	-	OverallWidth	Length	-	○	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
ドア	建具主寸法 H	ドア	寸法	高さ	1850mm	IfcDoor	-	-	OverallHeight	Length	-	○	-	-
ドア	ブラインドの有無	ドア	その他	ブラインドの有無	Yes	IfcDoor	-	Pset_DoorCommon	HasLouver※1	Boolean	-	○	-	-
部屋	(位置座標)	部屋	その他	定義済み IFC クラス	IfcSpace	IfcSpace	-	-	-	-	-	○	-	-
部屋	(位置座標)	部屋	その他	定義済み IFC タイプ			-	-	-	-	-	○	-	-
部屋	室名	部屋	識別情報	名前	事務室	IfcSpace	-	-	Name	Label	-	○	-	-
部屋	床面積	部屋	寸法	面積	96.00㎡	IfcSpace	-	Qto_SpaceBaseQuantities	NetPlannedArea	Area	-	○	-	-
部屋	所属階	部屋	拘束	レベル	1FL	IfcSpace	-	IfcRelContainedInSpatialStructureで リレーションを参照	IfcBuildingStorey.Name	Label	-	○	-	-
部屋	室用途	部屋	識別情報	省エネ法上 用途	事務室	IfcSpace	-	Pset_SpaceOccupancyRequirements	OccupancyType	Label	-	○	-	-
部屋	階高	部屋	寸法	部屋高さ(レ ベル指定)	4300mm	IfcSpace	-	Qto_SpaceBaseQuantities	Height	Length	-	○	-	-
部屋	天井高	部屋	識別情報	天井高	3.28m	IfcSpace	-	Qto_SpaceBaseQuantities	FinishCeilingHeight	Length	-	○	-	-
部屋	空調計算対象室である か否か	部屋	その他	空調計算対象室	Yes	IfcSpace	-	Pset_SpaceHVACDesign	AirConditioning ※空調機の有無	Boolean	-	○	-	-
部屋	換気計算対象室である か否か	部屋	その他	換気計算対象室	Yes	IfcSpace	-	Pset_SpaceHVACDesign	MechanicalVentilation ※機械換気の有無	Boolean	-	○	-	-

部位	情報	Revit 大分類	Revit 中分類	Revit 小分類	入力値例	IFC Entities	PredefinedType (* = USERDEFINED)	Property Set	Property Name	Property Type	材	エ	構	C
部屋	照明計算対象室であるか否か	部屋	その他	照明計算対象室	Yes	IfcSpace	-	Pset_SpaceLightingDesign	ArtificialLighting ※人工照明の有無	Boolean	-	○	-	-
部屋	給湯計算対象室であるか否か	部屋	その他	給湯計算対象室	Yes	IfcSpace	-	Pset_SpaceHVACDesign	HotWaterPresence ※2	Boolean	-	○	-	-
部屋	所属棟	部屋	文字	棟名称	-	IfcSpace	-	Pset_SpaceCommon	Building※3	Label	-	○	-	-
部屋	空調機群	部屋	その他	空調_室負荷 空調_外気負荷	PAC-1-1 PAC-1-1	IfcSpace	-	Pset_SpaceHVACDesign	AirHandlingName	Label	-	○	-	-
部屋	換気機器	部屋	その他	換気_給気 換気_排気 換気_循環 換気_空調	SF-6 EF-1	IfcSpace	-	Pset_SpaceHVACDesign	VentilationName※4	Label	-	○	-	-
部屋	照明器具名称	部屋	その他	照明_名称	FRS23-H321	IfcSpace	-	Pset_SpaceLightingDesign	LightingFixtures※5	Label	-	○	-	-
部屋	照明器具台数	部屋	その他	照明_台数	42	IfcSpace	-	Pset_SpaceLightingDesign	NumberOfLightingFixtures※6	Integer	-	○	-	-
部屋	節湯器具	部屋	その他	給湯_節湯	自動給湯栓	IfcSpace	-	Pset_SpaceHVACDesign	WaterSavingType※7	Label	-	○	-	-
部屋	給湯機器	部屋	その他	給湯_名称	EH-1	IfcSpace	-	Pset_SpaceHVACDesign	HotWaterType※8	Label	-	○	-	-

※1.HasLouver対応する情報項目が既存のスキーマにないため追加

※2.HotWaterPresence対応する情報項目が既存のスキーマにないため追加

※3.Building対応する情報項目が既存のスキーマにないため追加

※4.VentilationName対応する情報項目が既存のスキーマにないため追加

※5.LightingFixtures対応する情報項目が既存のスキーマにないため追加

※6.NumberOfLightingFixtures対応する情報項目が既存のスキーマにないため追加

※7.WaterSavingType対応する情報項目が既存のスキーマにないため追加

※8.HotWaterType対応する情報項目が既存のスキーマにないため追加

※9.Qto_ の数量セットで扱える情報項目の関係から、屋根であつてもIfcSlabを使用することが合理的であるとした

3.6 IFCパラメータマッピングテーブル

RevitにおけるIFC書き出し時のパラメータマッピングテーブルを以下のように設定しました。

```
#
PropertySet:    JPPset_TimberFrameDimension    I        IfcColumn, IfcColumnType
    Width    Length    b
    Height    Length    d
    StartingStorey    Text    基準レベル
    EndStorey    Text    上部レベル

#
PropertySet:    JPPset_TimberFrameDimension    I        IfcBeam, IfcBeamType
    Width    Length    b
    Height    Length    d
    StartingStorey    Text    参照レベル

#
PropertySet:    JPPset_TimberElementGeneral    I        IfcBeam, IfcBeamType, IfcColumn, IfcColumnType
    ProductName    Text    品名
    OriginalLength    Text    調達寸法
    TimberSpecies    Text    樹種名
    StrengthClass    Text    強度等級
    EnvironmentalCondition    Text    接着性能
    Adhesives    Text    使用接着剤
    SurfaceQuality    Text    材面の品質
    FormaldehydeEmissions    Text    ホルムアルデヒド放散量
    DryingProcess    Text    乾燥処理
    MoistureContent    Text    含水率
    PreservingMethod    Text    保存処理
    PreservationAgent    Text    木材保存剤
    ApprovedQuality    Text    AQ認証区分
    Manufacturer    Text    製造者名

#
PropertySet:    Qto_ColumnBaseQuantities    I        IfcColumn, IfcColumnType
    NetVolume    Volume    容積

#
PropertySet:    Qto_BeamBaseQuantities    I        IfcBeam, IfcBeamType
    NetVolume    Volume    容積

#
PropertySet:    JPPset_TimberElementGeneral    I        IfcWall, IfcWallType, IfcSlab, IfcSlabType
    ProductName    Text    品名
```

```

OriginalLength Text 調達寸法
TimberSpecies Text 樹種
StrengthClass Text 等級
EnvironmentalCondition Text 接着の性能
SurfaceQuality Text 板面の品質
FormaldehydeEmissions Text ホルムアルデヒド放散量
MoistureContent Text 含水率
PreservingMethod Text 保存処理
PreservationAgent Text 木材保存剤
ApprovedQuality Text AQ認証区分
Manufacturer Text 製造者名
InsectRepellent Text 防虫剤

#
PropertySet: JPPset_TimberConnection I IfcBeam, IfcBeamType, IfcColumn, IfcColumnType
ConnectionDetailsStarting Text StartJoint
ConnectionDetailsEnd Text EndJoint

#
PropertySet: Identity T IfcMaterial
Description Integer 建材番号

#
PropertySet: Pset_WallCommon T IfcWall, IfcWallType
isExternal Boolean 機能
ThermalTransmittance ThermalTransmittance 熱伝達係数(U)

#
PropertySet: Qto_WallBaseQuantities T IfcWall, IfcWallType
Width Length 幅

#
PropertySet: Qto_WallBaseQuantities I IfcWall, IfcWallType
NetSideArea Area 面積
Length Length 長さ
NetVolume Volume 容積

#
PropertySet: Qto_SlabBaseQuantities T IfcSlab, IfcSlabType
Depth Length 既定の厚さ

#
PropertySet: Pset_SlabCommon T IfcSlab, IfcSlabType
ThermalTransmittance ThermalTransmittance 熱伝達係数(U)

#
PropertySet: Qto_SlabBaseQuantities I IfcSlab, IfcSlabType
NetArea Area 面積
Length Length 長さ

```

```

NetVolume      Volume  容積

#
PropertySet:    Pset_WindowCommon      T      IfcWindow, IfcWindowType
    ThermalTransmittance      ThermalTransmittance      熱伝達係数 (U)
    HasLouver      Boolean  ブラインドの有無

#
PropertySet:    Pset_DoorCommon T      IfcDoor, IfcDoorType
    ThermalTransmittance      ThermalTransmittance      熱伝達係数 (U)
    HasLouver      Boolean  ブラインドの有無

#
PropertySet:    Pset_DoorWindowGlazingType      T      IfcDoor, IfcDoorType, IfcWindow, IfcWindowType
    SolarHeatGainTransmittance      ThermalTransmittance      日射熱取得率

#
PropertySet:    Qto_SpaceBaseQuantities I      IfcSpace
    NetPlannedArea      Area      面積
    Height      Length  部屋高さ(レベル指定)
    FinishCeilingHeight      Length  天井高

#
PropertySet:    Pset_SpaceCommon      I      IfcSpace
    Building      Text      棟名称

#
PropertySet:    Pset_SpaceOccupancyRequirements I      IfcSpace
    OccupancyType      Text      省エネ法上用途

#
PropertySet:    Pset_SpaceHVACDesign      I      IfcSpace
    AirConditioning Boolean  空調計算対象室
    MechanicalVentilation      Boolean  換気計算対象室
    HotWaterPresence      Boolean  給湯計算対象室
    AirHandlingName Text      空調_室負荷
    AirHandlingName Text      空調_外気負荷
    VentilationName Text      換気_給気
    VentilationName Text      換気_排気
    VentilationName Text      換気_循環
    VentilationName Text      換気_空調
    WaterSavingType Text      給湯_節湯
    HotWaterType      Text      給湯_名称

#
PropertySet:    Pset_SpaceLightingDesign      I      IfcSpace
    ArtificialLighting      Boolean  照明計算対象室
    LightingFixtures      Text      照明_名称

```

#	NumberOfLightingFixtures	Text	照明_台数
---	--------------------------	------	-------

4. IFCサンプルデータの活用方法

ここでは、前述した内容に基づき作成したIFCデータを用いて、一部のユースケースに活用していく実践例を示しています。

4.1 炭素貯蔵量の算定

構造IFCデータから、柱と梁の情報を抽出する作業例です。付属のPythonプログラムを参考に実行します。

また、抽出するプロパティは「建築物に利用した木材に係る炭素貯蔵量の算定シート」(※)におけるデータ行と一致させています。※掲載先: 林野庁ウェブサイト「建築物に利用した木材に係る炭素貯蔵量の表示に関するガイドライン」<https://www.rinya.maff.go.jp/j/mokusan/mieruka.html>

4.1.1 出力フォーマット

[設定]

```
FUNC extract_beams:
    ENTITY IfcBeam
    ATTRIBUTE PredefinedType
    PROPERTIES Pset_BeamCommon.Reference, Qto_BeamBaseQuantities.NetVolume
    TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName, JPPset_TimberElementGeneral.StrengthClass, JPPset_TimberElementGeneral.TimberSpecies
    SPATIAL IfcBuildingStorey.Name
    MATERIAL
    FILTER TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName == 'JAS構造用集成材' or TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName == 'JAS機械等級区分構造用集成材'
END_FUNC
```

```
FUNC extract_columns:
    ENTITY IfcColumn
    ATTRIBUTE PredefinedType
    PROPERTIES Pset_ColumnCommon.Reference, Qto_ColumnBaseQuantities.NetVolume
    TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName, JPPset_TimberElementGeneral.StrengthClass, JPPset_TimberElementGeneral.TimberSpecies
    SPATIAL IfcBuildingStorey.Name
    MATERIAL
    FILTER TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName == 'JAS構造用集成材' or TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName == 'JAS機械等級区分構造用集成材'
END_FUNC
```

[出力]

製材区分（製材・集成材・C L T等）, 構造フレーム, extract_beams(JPPset_TimberElementGeneral.TimberSpecies), [SUBCOUNT:Qto_BeamBaseQuantities.NetVolume]

製材区分（製材・集成材・C L T等）, 構造柱, extract_columns(JPPset_TimberElementGeneral.TimberSpecies), [SUBCOUNT:Qto_ColumnBaseQuantities.NetVolume]

4.1.2 出力結果

出力結果は以下の通り。ここでは、シートの体裁通り、区分や樹種ごとの木材使用量が算定されました。

```
製材区分（製材・集成材・CLT等）,構造フレーム,カラマツ,87.128496000000129
製材区分（製材・集成材・CLT等）,構造フレーム,スギ,26.633880000000694
製材区分（製材・集成材・CLT等）,構造フレーム,ヒノキ,5.403762000000095
製材区分（製材・集成材・CLT等）,構造柱,カラマツ,0.16096499999999997
製材区分（製材・集成材・CLT等）,構造柱,ヒノキ,13.630392
```

4.1.3 情報連携

このテキストデータを、「建築物に利用した木材に係る炭素貯蔵量の算定シート」のエクセルファイルにインポートします。

シートを開き、データ>データの取得と変換>データの取得>ファイルから>テキストまたはcsvからを選択し、出力結果のテキストファイルを選択します。

読み込み先...>[テーブル],「既存のワークシート, [= '1_入力シート'!\$C\$22]」>OKをクリックすると、入力シートのセル内に読み込まれるようになります。この時先頭行がヘッダーとして扱われエラーを吐くため、テーブル デザイン>テーブル スタイルのオプション>[見出し行]のチェックを外します。

No	区分	部材、製品名等	建築物に利用した木材の量 [単位:m]							木材の炭素含有率	建築物に利用した炭素貯蔵量 [単位:t-CO ₂]	産地別のCO ₂ 削減率	
			国産材			国産材以外(国産材・国産材以外の区分不明を含む)			合計 [単位:m]				
			樹種	利用量 [単位:m]	木材の密度 [単位:t/m]	樹種	利用量 [単位:m]	木材の密度 [単位:t/m]				国産材 [単位:t-CO ₂]	国産材以外 [単位:t-CO ₂]
1													
2	製材区分(製材・集成材・CLT等)	構造柱	カラマツ	0.160965	0.435				0.2	0.500	0.1	0.1	
3	製材区分(製材・集成材・CLT等)	構造柱	ヒノキ	13.630392	0.383				13.6	0.500	9.6	9.6	
4	製材区分(製材・集成材・CLT等)	構造フレーム	カラマツ	87.128496	0.435				87.1	0.500	69.5	69.5	
5	製材区分(製材・集成材・CLT等)	造作フレーム	スギ	26.63388	0.331				26.6	0.500	16.1	16.1	
6	製材区分(製材・集成材・CLT等)	構造フレーム	ヒノキ	5.403762	0.383				5.4	0.500	3.8	3.8	

出力シートに算定結果が表示されました。

■ 建築物に利用した木材の炭素貯蔵量の表示例

建物名（住所）に利用した木材に係る炭素貯蔵量（CO₂換算）

延べ床面積	国産材 利用量	国産材の 炭素貯蔵量 (CO ₂ 換算)	木材全体 利用量	木材全体の 炭素貯蔵量 (CO ₂ 換算)
0	133	99	133	99
m ²	m ³	t-CO ₂	m ³	t-CO ₂

この表示は、林野庁「建築物に利用した木材の炭素貯蔵量の表示ガイドライン」（令和3年10月1日付け3林政産第85号林野庁長官通知）に準拠し、この建築物に利用し

4.2 材料調達連携

構造IFCデータから、柱と梁(土台は除く)の情報を抽出する作業例です。付属のPythonプログラムを参考に実行します。
また、抽出するプロパティは木材製品供給情報データベース(「もりんく」)における木質材料に関する登録項目と一致させています。

4.2.1 テンプレートファイルの設定

[設定]

```
FUNC extract_beams:
    ENTITY IfcBeam
    ATTRIBUTE PredefinedType
    PROPERTIES Pset_BeamCommon.Reference, Pset_BeamCommon.Span
    TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName, JPPset_TimberElementGeneral.StrengthClass, JPPset_TimberElementGeneral.TimberSpecies
    SPATIAL IfcBuildingStorey.Name
    MATERIAL
    FILTER PredefinedType == 'BEAM'
END_FUNC

FUNC extract_columns:
    ENTITY IfcColumn
    ATTRIBUTE PredefinedType
    PROPERTIES Pset_ColumnCommon.Reference, JPPset_TimberFrameDimension.Length
    TYPEPROPERTIES JPPset_TimberElementGeneral.ProductName, JPPset_TimberElementGeneral.StrengthClass, JPPset_TimberElementGeneral.TimberSpecies
    SPATIAL IfcBuildingStorey.Name
    MATERIAL
    FILTER PredefinedType == 'COLUMN'
END_FUNC
```

[出力]

```
extract_beams(Pset_BeamCommon.Reference), (Pset_BeamCommon.Span), (JPPset_TimberElementGeneral.ProductName), (JPPset_TimberElementGeneral.TimberSpecies), (JPPset_TimberElementGeneral.StrengthClass)
extract_columns(Pset_ColumnCommon.Reference), (JPPset_TimberFrameDimension.Length), (JPPset_TimberElementGeneral.ProductName), (JPPset_TimberElementGeneral.TimberSpecies), (JPPset_TimberElementGeneral.StrengthClass)
```

4.2.2 出力結果

出力結果は以下の通りとなります。ここでは、もりんくのJAS等構造材検索で要求される選択項目に合わせ、断面寸法や設計長さ、樹種毎等に本数を出力しました。

```
240x840,5460.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),27
210x600,5460.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),23
210x840,10920.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),12
```

210x840,6370.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),6
210x840,5460.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),15
180x600,7280.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),30
210x840,10010.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),77
180x600_E75,7280.00,JAS構造用集成材,スギ,E75-F240(異等級構成),18
150x600_E75,5460.00,JAS構造用集成材,スギ,E75-F240(異等級構成),15
180x600_E75,5460.00,JAS構造用集成材,スギ,E75-F240(異等級構成),6
150x450_E75,5460.00,JAS構造用集成材,スギ,E75-F240(異等級構成),9
120x240,2730.00,JAS機械等級区分構造用製材,ヒノキ,甲種2級,30
120x210,2730.00,JAS機械等級区分構造用製材,スギ,E70,75
120x270,2730.00,JAS機械等級区分構造用製材,ヒノキ,E70,15
120x180,1820.00,JAS機械等級区分構造用製材,スギ,E70,1136
210x600,5160.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),2
210x600,5300.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),2
240x840,1820.00,JAS構造用集成材,カラマツ,E105-F300(異等級構成),1
210 x 210,2810.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),2
210 x 120,2810.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),18
240 x 210,2810.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),2
210 x 120,3050.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),20
210 x 180,3050.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),6
210 x 150,3050.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),2
240 x120,2810.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),22
240 x120,3230.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),4
210 x 210,3230.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),4
210 x 180,2810.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),2
180 x 150,3050.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),6
180 x 120,3050.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),12
150 x 150,3050.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),13
210 x 150,2810.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),14
150 x 150,3200.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),7
210 x 210,2860.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),12
210 x 120,2860.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),36
240 x 210,2860.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),4
210 x 120,3100.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),38
210 x 210_カラマツ,3100.00,JAS構造用集成材,カラマツ,E95-F315(同一等級),1
210 x 180,3100.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),13
210 x 150,3100.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),4
240 x120,2860.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),52
210 x 180,2860.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),4
180 x 150,3100.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),14
180 x 120,3100.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),24

150 x 150,3100.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),24
210 x 150,2860.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),27
150 x 150,3250.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),14
210 x 150,3700.00,JAS構造用集成材,ヒノキ,E95-F315(同一等級),1

4.2.3 情報連携

出力した材料の最大寸法等をもとに、その寸法に対応できる事業者（プレカット加工業等）を検索することができます。

JAS等構造材から選ぶ

JAS等構造材品名JAS構造用集成材

短辺 (mm)

☐ 全て選択

☐ 105☐ 120☐ 150☐ 180☒ 210☐ 89☐ 140☐ その他

長辺 (mm)

☐ 全て選択

☐ 105☐ 140☐ 120☐ 184☐ 150☐ 235☐ 180☐ 286☐ 210☐ 336☐ 240☐ 387☐ 270☐ 300☐ 330☐ 360☐ 390☐ 420☐ 450☐ 480☐ 510☐ 540☐ 570☐ 600☐ 630☐ 660☐ 690☐ 720☐ 750☐ 780☐ 810☒ 840☐ 870☐ 900☐ その他

材長 (mm)

☐ 全て選択

☐ 1910☐ 2700☐ 2750☐ 2870☐ 2850☐ 3830☐ 2980☐ 4690☐ 2985☐ 5750☐ 3000☐ 6004☐ 3650☐ その他☐ 3985☐ 4000☐ 4600☐ 4980☐ 5000☐ 5980☐ 5985☐ 6000☐ 6500☐ 7000☐ 7500☒ 8000☐ 8500☐ 9000☐ 9500☐ 10000☐ 10500☐ 11000☐ 11500☐ 12000

樹種名

☐ 全て選択

☐ スギ☐ ヒノキ☐ カラマツ☐ トドマツ☐ オウシュウアカマツ☐ スプルース☐ ベイマツ☐ ダフリカカラマツ☐ サザンパイン☐ 混合(カラマツ+スギ)☐ 混合(ヒノキ+スギ)

見つからない場合は、個別に問い合わせるか、構造計画の変更を行います。

また、可能であっても納期の対応が難しい場合も考えられますので、先ほど出力したテキストファイル(CSV化したもの等)を個別に問い合わせに活用すると効果的です。

5. 参考プログラム

5.1 Pythonプログラムとテンプレートファイルの使用例

コマンドラインから次のように実行します。

```
python ifcextract.py <IFCファイル> <テンプレートファイル>
```

実行すると、テンプレートファイルに定義された設定に基づいてIFCファイルからデータを抽出し、定義された出力設定に基づき、結果をoutput.txtとして出力します。

テンプレートファイルの記述方法

テンプレートファイルは、本プログラム処理のための固有の記述方法としており、[設定] セクションと[出力] セクションに分かれています。

[設定] セクション

設定セクションでは、各関数ブロック(FUNC ... END_FUNC)内に抽出条件を記述します。各ブロックの内容は以下のような項目で構成されます。

- FUNC 関数名:
 - 任意の関数名を指定します。出力テンプレート内でこの名前を参照して抽出処理を行います。
- ENTITY:
 - 対象となる IFC エンティティの種類(例: IfcBeam, IfcWall など)を指定します。
- ATTRIBUTE:
 - IFC エンティティの直接属性(例: GlobalId, Name など)をカンマ区切りで指定します。
- PROPERTIES:
 - IFC エンティティに関連付けられたプロパティセットから抽出する情報を指定します。形式は プロパティセット名.プロパティ名 または単に プロパティ名 とします。
- TYPEPROPERTIES:
 - IFC エンティティのタイプオブジェクト(例: IfcBeamType)に定義されているプロパティを抽出する場合に使用します。形式は PROPERTIES と同様です。
- SPATIAL:
 - IFC エンティティが属する空間情報を抽出します。

- 例: IfcBuildingStorey.Name と指定すると、エンティティの ContainedInStructure リレーションから該当する空間の Name を取得します。
- MATERIAL:
 - 材料情報を抽出するための指定です。
 - 例: IfcMaterialLayer.Name とすると、材料情報から名前を取得します。
- FILTER:
 - 抽出対象のエンティティをフィルタリングするための条件を指定します。
 - 複数条件の場合、and および or を使用できます。
 - また、条件の先頭に以下のキーワードを付けることで、対象の情報を明示的に指定できます:
 - SPATIAL : 空間情報 (get_spatial_value で取得) に対して条件を適用
 - PROPERTIES : インスタンスレベルのプロパティに対して条件を適用
 - TYPEPROPERTIES : タイプオブジェクトのプロパティに対して条件を適用
 - MATERIAL : 材料情報に対して条件を適用
 - 何も付けない場合は、直接属性として評価します。

[設定]

```
FUNC extract_beams:
  ENTITY IfcBeam
  ATTRIBUTE GlobalId, Name, PredefinedType
  PROPERTIES Pset_BeamCommon.Reference, Pset_BeamCommon.Span
  TYPEPROPERTIES
  SPATIAL IfcBuildingStorey.Name
  MATERIAL IfcMaterialLayer.Name
  FILTER SPATIAL IfcBuildingStorey.Name == '2FL' and PredefinedType == 'BEAM'
END_FUNC
```

上記の例では、IfcBeam エンティティから以下の情報を抽出します:

- 直接属性: GlobalId, Name, PredefinedType
- インスタンスレベルのプロパティ: Pset_BeamCommon.Reference, Pset_BeamCommon.Span
- タイプオブジェクトのプロパティ: 指定しない
- 空間情報: IfcBuildingStorey.Name
- 材料情報: IfcMaterialLayer.Name
- また、これらはフィルタリング条件により、空間情報が '2FL' で、かつ直接属性 PredefinedType が 'BEAM' のエンティティのみを対象とします。

[出力] セクション

出力セクションでは、抽出結果の出力フォーマットを指定します。各行は、関数名とその後に丸括弧や角括弧で抽出フィールドを指定する形式です。

- レコードレベル出力:
 - 各エンティティごとに指定したフィールドの値を出力します。この時、丸括弧 () で指定します。また、[SUBCOUNT]と記述すると、対象フィールドの組み合わせごとにグループ化し、そのグループに含まれる要素数を出力します。`[SUBCOUNT:\*\*\*]`と記述すると、その値を小計します。

```
extract_beams(GlobalId), (Name), (Pset_BeamCommon.Reference), (IfcBuildingStorey.Name)
extract_beams(GlobalId), (Name), (Pset_BeamCommon.Reference), (IfcBuildingStorey.Name), [SUBCOUNT]
```

この例では、extract_beams内の各IfcBeamの GlobalId、Name、プロパティ Pset_BeamCommon.Reference、空間情報 をカンマ区切りで出力します。

- 集計出力:
 - すべてのエンティティの指定フィールド(数値型に限ります)を合計し出力します。この時、角括弧 [] で指定します。リテラル文字列を付加して出力することも可能です。また、[COUNT]と記述すると、要素の個数を返します。

```
extract_beams[Pset_BeamCommon.Span], 固定値, [COUNT]
```

この例では、extract_beams内の全Ifcbeamの Pset_BeamCommon.Span の数値を合計し、結果の後に 固定値 という文字列を追加し、さらに要素個数を出力します。

Pythonプログラムの例

以下はIFCデータから情報を抽出するためのPyhtonプログラムの例です。また、本書に掲載されているコードは参考用です。本コードを実行したことによるいかなる損害や不具合についても、著者および関係者は一切の責任を負いません。ご利用は自己責任でお願いいたします。

```
import sys
import re
import ifcopenshell

def parse_template(template_path):
    functions = {}
    output_templates = []
    with open(template_path, encoding="utf-8") as f:
        lines = [line.strip() for line in f if line.strip()]
    i = 0
    while i < len(lines):
        line = lines[i]
        if line == "[設定]":
```

```

i += 1
while i < len(lines) and not lines[i].startswith("["):
    if lines[i].startswith("FUNC"):
        m = re.match(r'FUNC\s+(\w+):', lines[i])
        if m:
            func_name = m.group(1)
            func_def = {
                "entity": None,
                "attributes": [],
                "properties": [],
                "type_properties": [],
                "spatial": None,
                "material": None,
                "filter": None
            }
            i += 1
            while i < len(lines) and lines[i] != "END_FUNC":
                cur = lines[i]
                if cur.startswith("ENTITY"):
                    parts = cur.split()
                    if len(parts) >= 2:
                        func_def["entity"] = parts[1]
                elif cur.startswith("ATTRIBUTE"):
                    func_def["attributes"] = [x.strip() for x in cur[len("ATTRIBUTE"):].strip().split(",")]
                elif cur.startswith("PROPERTIES"):
                    func_def["properties"] = [x.strip() for x in cur[len("PROPERTIES"):].strip().split(",")]
                elif cur.startswith("TYPEPROPERTIES"):
                    func_def["type_properties"] = [x.strip() for x in cur[len("TYPEPROPERTIES"):].strip().split(",")]
                elif cur.startswith("SPATIAL"):
                    func_def["spatial"] = cur[len("SPATIAL"):].strip()
                elif cur.startswith("MATERIAL"):
                    func_def["material"] = cur[len("MATERIAL"):].strip()
                elif cur.startswith("FILTER"):
                    func_def["filter"] = cur[len("FILTER"):].strip()
                i += 1
            i += 1 # END_FUNC
            functions[func_name] = func_def
        else:
            i += 1
    else:
        i += 1

```



```

elif line == "[出力]":
    i += 1
    while i < len(lines) and not lines[i].startswith("["):
        output_templates.append(lines[i])
        i += 1
    else:
        i += 1
return functions, output_templates

def get_spatial_value(entity, spatial_spec):
    parts = spatial_spec.split(".")
    if len(parts) != 2:
        return None
    target_entity, target_attr = parts
    if hasattr(entity, "ContainedInStructure"):
        for rel in entity.ContainedInStructure:
            struct = rel.RelatingStructure
            if struct.is_a(target_entity):
                return getattr(struct, target_attr, None)
    return None

def get_property_value(source, property_token):
    if '.' in property_token:
        set_name, prop_name = [x.strip() for x in property_token.split(".")]
    else:
        set_name, prop_name = None, property_token.strip()
    rels = []
    if hasattr(source, "IsDefinedBy") and source.IsDefinedBy:
        rels = [rel.RelatingPropertyDefinition for rel in source.IsDefinedBy if rel.RelatingPropertyDefinition.is_a("IfcPropertySet")]
    elif hasattr(source, "HasPropertySets") and source.HasPropertySets:
        rels = [pset for pset in source.HasPropertySets if pset.is_a("IfcPropertySet")]
    else:
        return None
    for pset in rels:
        if set_name and getattr(pset, "Name", "").strip().lower() != set_name.lower():
            continue
        if hasattr(pset, "HasProperties"):
            for prop in pset.HasProperties:
                if getattr(prop, "Name", "").strip().lower() == prop_name.lower():
                    if hasattr(prop, "NominalValue"):
                        return prop.NominalValue.wrappedValue if hasattr(prop.NominalValue, "wrappedValue") else prop.NominalValue

```

```

return None

def get_type_property_value(entity, property_token):
    if not hasattr(entity, "IsTypedBy"):
        return None
    for rel in entity.IsTypedBy:
        type_obj = rel.RelatingType
        if type_obj:
            value = get_property_value(type_obj, property_token)
            if value is not None:
                return value
    return None

def get_material_name(entity):
    if not hasattr(entity, "HasAssociations"):
        return None
    for rel in entity.HasAssociations:
        if rel.is_a("IfcRelAssociatesMaterial"):
            mat = rel.RelatingMaterial
            if mat.is_a("IfcMaterial"):
                return getattr(mat, "Name", None)
            elif mat.is_a("IfcMaterialLayerSetUsage"):
                layer_set = getattr(mat, "ForLayerSet", None)
                if layer_set and hasattr(layer_set, "MaterialLayers") and layer_set.MaterialLayers:
                    return getattr(layer_set.MaterialLayers[0], "Name", None)
            elif mat.is_a("IfcMaterialList"):
                names = [getattr(m, "Name", None) for m in mat.Materials if getattr(m, "Name", None)]
                return ", ".join(names)
    return None

def evaluate_atomic_condition(entity, spatial_value, cond):
    cond = cond.strip()
    if not cond:
        return True
    # SPATIAL条件
    if cond.lower().startswith("spatial"):
        cond = cond[len("spatial"):].strip()
        m = re.match(r'([^\s]+)\s*=\s*([^\s]+)', cond)
        if not m:
            return False
        expected = m.group(2).strip()

```

```

    if spatial_value is None:
        return False
    if hasattr(spatial_value, "wrappedValue"):
        spatial_value = spatial_value.wrappedValue
    return str(spatial_value) == expected
# PROPERTIES条件
elif cond.lower().startswith("properties"):
    cond = cond[len("properties"):].strip()
    m = re.match(r'([^\s]+)==s*\'([^\s]+)\'', cond)
    if not m:
        return False
    token = m.group(1).strip()
    expected = m.group(2).strip()
    actual = get_property_value(entity, token)
    if actual is None:
        return False
    if hasattr(actual, "wrappedValue"):
        actual = actual.wrappedValue
    return str(actual) == expected
# TYPEPROPERTIES条件
elif cond.lower().startswith("typeproperties"):
    cond = cond[len("typeproperties"):].strip()
    m = re.match(r'([^\s]+)==s*\'([^\s]+)\'', cond)
    if not m:
        return False
    token = m.group(1).strip()
    expected = m.group(2).strip()
    actual = get_type_property_value(entity, token)
    if actual is None:
        return False
    if hasattr(actual, "wrappedValue"):
        actual = actual.wrappedValue
    return str(actual) == expected
# MATERIAL条件
elif cond.lower().startswith("material"):
    cond = cond[len("material"):].strip()
    m = re.match(r'([^\s]+)==s*\'([^\s]+)\'', cond)
    if not m:
        return False
    expected = m.group(2).strip()
    actual = get_material_name(entity)

```

```

    if actual is None:
        return False
    if hasattr(actual, "wrappedValue"):
        actual = actual.wrappedValue
    return str(actual) == expected
# 直接属性条件
else:
    m = re.match(r'([^\s]+)==\s*\'([^\s]+)\'', cond)
    if not m:
        return False
    field = m.group(1).strip()
    expected = m.group(2).strip()
    actual = getattr(entity, field, None)
    if actual is None:
        return False
    if hasattr(actual, "wrappedValue"):
        actual = actual.wrappedValue
    return str(actual) == expected

def evaluate_filter_expression(entity, spatial_value, filter_expr):
    or_parts = [part.strip() for part in re.split(r'\s+or\s+', filter_expr, flags=re.IGNORECASE) if part.strip()]
    for part in or_parts:
        and_parts = [cond.strip() for cond in re.split(r'\s+and\s+', part, flags=re.IGNORECASE) if cond.strip()]
        if all(evaluate_atomic_condition(entity, spatial_value, cond) for cond in and_parts):
            return True
    return False

def apply_filter(entity, func_def, spatial_value):
    filter_str = func_def.get("filter", "")
    if not filter_str:
        return True
    return evaluate_filter_expression(entity, spatial_value, filter_str)

def extract_entities(ifc_file, func_def):
    records = []
    entities = ifc_file.by_type(func_def["entity"])
    for entity in entities:
        spatial_value = get_spatial_value(entity, func_def["spatial"]) if func_def.get("spatial") else None
        if not apply_filter(entity, func_def, spatial_value):
            continue
        record = {}

```

```

    for attr in func_def.get("attributes", []):
        record[attr] = getattr(entity, attr, None)
    for prop in func_def.get("properties", []):
        record[prop] = get_property_value(entity, prop)
    for tprop in func_def.get("type_properties", []):
        record[tprop] = get_type_property_value(entity, tprop)
    if func_def.get("spatial"):
        record[func_def["spatial"]] = spatial_value
    if func_def.get("material"):
        record[func_def["material"]] = get_material_name(entity)
    records.append(record)
return records

def process_output_template(template_line, records):
    tokens = [token.strip() for token in template_line.split(",")]
    # SUBCOUNT 部分の判定
    subcount = False
    subcount_field = None
    new_tokens = []
    for token in tokens:
        token_up = token.upper()
        m_sub = re.match(r'\[SUBCOUNT(?:\s*([^\]]+))?\]', token, flags=re.IGNORECASE)
        if m_sub:
            subcount = True
            subcount_field = m_sub.group(1) # Noneの場合は単純カウント
        else:
            new_tokens.append(token)
    tokens = new_tokens

    output_lines = []
    # レコードレベル出力（丸括弧指定）の場合
    if any(re.search(r'\w+\(', token) or (token.startswith("(") and token.endswith(")")) for token in tokens):
        if subcount:
            # 各レコードから出力行を生成し、グループ化してカウントまたは数値合計
            group_dict = {}
            for rec in records:
                out_fields = []
                for token in tokens:
                    m = re.search(r'(\w+)\s*([^\)]+)\s*', token)
                    if m:
                        field = m.group(2).strip()

```

```

        value = rec.get(field, "")
        if isinstance(value, (int, float)):
            out_fields.append(f"{value:.2f}")
        else:
            out_fields.append(str(value))
    elif token.startswith("(") and token.endswith(")"):
        field = token[1:-1].strip()
        value = rec.get(field, "")
        if isinstance(value, (int, float)):
            out_fields.append(f"{value:.2f}")
        else:
            out_fields.append(str(value))
    else:
        if token:
            out_fields.append(token)
    key = tuple(out_fields)
    if key not in group_dict:
        if subcount_field:
            group_dict[key] = 0.0
        else:
            group_dict[key] = 0
    if subcount_field:
        try:
            group_dict[key] += float(rec.get(subcount_field, 0))
        except Exception:
            pass
    else:
        group_dict[key] += 1
for key, total in group_dict.items():
    if subcount_field:
        output_lines.append(",".join(list(key) + [f"{total:.2f}"]))
    else:
        output_lines.append(",".join(list(key) + [str(total)]))
else:
    for rec in records:
        out_fields = []
        for token in tokens:
            m = re.search(r'(\w+)\(((\[^\])+)\)', token)
            if m:
                field = m.group(2).strip()
                value = rec.get(field, "")

```

```

        if isinstance(value, (int, float)):
            out_fields.append(f"{value:.2f}")
        else:
            out_fields.append(str(value))
    elif token.startswith("(") and token.endswith(")"):
        field = token[1:-1].strip()
        value = rec.get(field, "")
        if isinstance(value, (int, float)):
            out_fields.append(f"{value:.2f}")
        else:
            out_fields.append(str(value))
    else:
        if token:
            out_fields.append(token)
        output_lines.append(",".join(out_fields))

# 集計出力 (角括弧指定) の場合
elif any(re.search(r'\w+\[', token) or token.startswith("[") for token in tokens):
    out_fields = []
    for token in tokens:
        if token.upper() == "[COUNT]":
            out_fields.append(str(len(records)))
        else:
            m = re.search(r'(\w+)\[([^\]]+)\]', token)
            if m:
                field = m.group(2).strip()
                total = sum(float(rec.get(field, 0)) for rec in records if rec.get(field, "") != "")
                out_fields.append(f"{total:.2f}")
            else:
                if token:
                    out_fields.append(token)
        output_lines.append(",".join(out_fields))
    return output_lines

def main():
    if len(sys.argv) < 3:
        print("Usage: python ifcextract.py <ifc_file> <template_file>")
        sys.exit(1)
    ifc_path = sys.argv[1]
    template_path = sys.argv[2]
    functions, output_templates = parse_template(template_path)
    if not functions:

```

```

    print("設定セクションに関数定義がありません。")
    sys.exit(1)

try:
    ifc_file = ifcopenshell.open(ifc_path)
except Exception as e:
    print(f"IFC ファイルのオープンに失敗しました: {e}")
    sys.exit(1)

all_output = []
for tpl in output_templates:
    m = re.search(r'(\w+) [\(\[\'', tpl)
    if not m:
        print("関数名がテンプレートから抽出できません:", tpl)
        continue
    func_name = m.group(1)
    if func_name not in functions:
        print(f"テンプレート内に {func_name} 関数が定義されていません。")
        continue
    func_def = functions[func_name]
    records = extract_entities(ifc_file, func_def)
    outs = process_output_template(tpl, records)
    all_output.extend(outs)

with open("output.txt", "w", encoding="utf-8") as f:
    for line in all_output:
        f.write(line + "\n")

print("出力は output.txt に保存されました。")

if __name__ == "__main__":
    main()

```