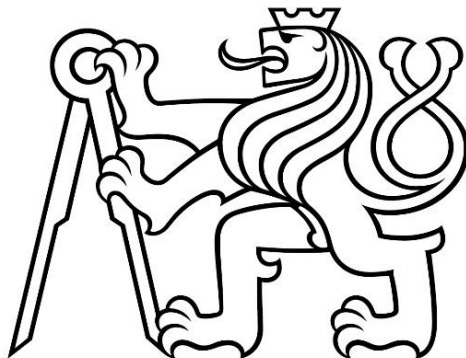


České vysoké učení technické v Praze  
Fakulta stavební  
Katedra konstrukcí pozemních staveb

Diplomová práce  
Využití systému BIM pro Demontovatelný  
prefabrikovaný systém



Autor: Střelka Vít, Bc.

Vedoucí práce: Ing. Radek Zigler, Ph.D.

Datum odevzdání: 2.1.2022

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Střelka** Jméno: **Vít** Osobní číslo: **460371**  
Fakulta/ústav: **Fakulta stavební**  
Zadávací katedra/ústav: **Katedra konstrukcí pozemních staveb**  
Studijní program: **Stavební inženýrství**  
Studijní obor: **Konstrukce pozemních staveb**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Využití systému BIM pro Demontovatelný prefabrikovaný systém**

Název diplomové práce anglicky:

**Use of the BIM system for a demountable prefabricated system**

Pokyny pro vypracování:

Pro systém DPS (demontovatelný stavební systém) vytvořte v softwaru Revit knihovnu dynamicky generovaných parametrických prvků. Vytvořte plugin založený na Dynamo scriptu do softwaru Revit pro generování typových prvků z prototypové knihovny s možností přímého zadání parametrů uživatelem. S využitím těchto prvků vytvořte komplexní BIM model ukázkové konstrukce, včetně analytického statického modelu a jeho následného využití v konečněprvkovém softwaru Autodesk Robot, resp. Scia Engineer.

Seznam doporučené literatury:

Manuál Dynamo Primer - <https://primer.dynamobim.org>  
DataShapes rozšíření - <https://data-shapes.io>  
Manuál k programovacímu jazyku Python pro Revit - <https://docs.python.org/3/tutorial>  
BIM příručka - <http://issuu.com/czbim/docs/bim-priručka-2013-v1>  
Revit ve stavební praxi - [http://issuu.com/oktaedr/docs/oktaedr\\_revit\\_ve\\_stavebni\\_praxi](http://issuu.com/oktaedr/docs/oktaedr_revit_ve_stavebni_praxi)

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Radek Zigler, Ph.D., katedra konstrukcí pozemních staveb FSv**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **29.09.2021**

Termín odevzdání diplomové práce: **02.01.2022**

Platnost zadání diplomové práce: \_\_\_\_\_

Ing. Radek Zigler, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Jiří Máca, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_ Datum převzetí zadání

\_\_\_\_\_ Podpis studenta

# SPECIFIKACE ZADÁNÍ

Jméno diplomanta: Střelka Vít

Název diplomové práce: Využití systému BIM pro Demontovatelný prefabrikovaný systém

Základní část: KPS podíl: 100 %

Formulace úkolů: Pro systém DPS (demontovatelný stavební systém) vytvořte v softwaru Revit knihovnu dynamicky generovaných parametrických prvků. Vytvořte plugin založený na Dynamo scriptu do softwaru Revit pro generování typových prvků z prototypové knihovny s možností přímého zadání parametrů uživatelem. S využitím těchto prvků vytvořte komplexní BIM model ukázkové konstrukce, včetně analytického statického modelu a jeho následného využití v konečněprvkovém softwaru Autodesk Robot, resp. Scia Engineer.

Podpis vedoucího DP: ..... Datum: .....

Případné další části diplomové práce (části a jejich podíl určí vedoucí DP):

2. Část: ..... podíl: ..... %

Konzultant (jméno, katedra): .....

Formulace úkolů: .....

Podpis konzultanta: ..... Datum: .....

3. Část: ..... podíl: ..... %

Konzultant (jméno, katedra): .....

Formulace úkolů: .....

Podpis konzultanta: ..... Datum: .....

4. Část: ..... podíl: ..... %

Konzultant (jméno, katedra): .....

Formulace úkolů: .....

Podpis konzultanta: ..... Datum: .....

Poznámka:

Zadání včetně vyplněných specifikací je nedílnou součástí diplomové práce a musí být přiloženo k odevzdané práci. (Vyplněné specifikace není nutné odevzdat na studijní oddělení spolu s 1. stranou zadání již ve 2. týdnu semestru)

## **Čestné prohlášení:**

Prohlašuji, že jsem diplomovou práci na téma využití systému BIM na Demontovatelný prefabrikovaný systém vypracoval samostatně a s použitím uvedené literatury, softwarů a zdrojů v souladu s metodickým pokynem o etické přípravě vysokoškolských a závěrečných prací.

V ..... dne .....

.....

Střelka Vít, Bc.

## **Poděkování:**

Tímto bych chtěl poděkovat vedoucímu závěrečné práce panu Ing. Radku Ziglerovi, Ph.D. za odborné vedení, lidský přístup a pomoc při vypracování diplomové práce. Dále bych chtěl poděkovat panu Ing. Jakubovi Veselce za rady týkající se modelování prvků v Revitu. Jsem vděčný za všechny rady, typy a podklady, které mi výrazně pomohly k vypracování této práce. Zároveň bych chtěl poděkovat všem, kteří mě při zpracování práce podporovali, hlavně rodině a mým nejbližším přátelům.

## **Abstrakt:**

Diplomová práce se zabývá aplikováním BIMu na Demontovatelný prefabrikovaný systém, dále jen DPS. Tento systém vyvinutý na Fakultě stavební ČVUT byl v rámci diplomové práce parametrizován a představen na vzorovém modelu fiktivní stavby. Současně byl vytvořen skript pro software Revit, který zajišťuje jednoduší integraci systému do modelu uživatele, nebo projektanta. Na prvcích systému DPS byly předvedeny výhody BIM modelu, jako databáze dat, z hlediska tvorby dokumentace a například i znovupoužití prvků. Model byl také převeden do statického softwaru Robot Structural Analysis pro statickou analýzu a následně do softwaru Navisworks Manage, pro přípravu harmonogramu a animace výstavby.

## **Klíčová slova:**

BIM, Informační model stavby, Informační modelování stavby, Revit, Navisworks, Dynamické prvky, Parametrizace, DPS, Demontovatelný prefabrikovaný systém, 3D Model, Dynamo, Vizuální programování, Statická analýza, Analytický model, Harmonogram, Schéma výstavby

## **Abstract:**

Master thesis deals with the application of BIM to the prefabricated demountable system called DPS. This system developed at the Faculty of Civil Engineering at the Czech Technical University in Prague was parameterized within the master thesis and presented on a model of a fictitious building. At the same time, a script for Revit software was created, which ensures easier integration of the system into the users model. The advantages of the BIM model as data database, were demonstrated on the creation of documentation and for presentation on reuse of the elements. The model was also transferred to the static software Robot Structural Analysis, for static analysis and subsequently to the software Navisworks Manage, for the preparation of the construction schedule and animation.

## **Key words:**

BIM, Building information model, Bulding information modeling, Revit, Navisworks, Dynamic elements, Parametrization, DPS, Demountable prefabricated systém, 3D Model, Dynamo, Visual programming, Static analysis, Analytical Model, Construction schedule, Construction scheme

## Obsah

1. Cíl práce.....	10
1.1. Použité pojmy a zkratky.....	10
2. Úvod do informačního modelování .....	11
2.1. BIM a implementace v rámci ČR a Evropy.....	11
2.2. Digitální dvojče jako databáze informací .....	11
2.3. Revit, nástroj na tvorbu BIM modelu .....	12
3. Systém DPS s využitím BIMu .....	13
3.1. Parametrické navrhování .....	13
3.2. Obecné modelování prvků systému DPS.....	18
3.3. Využití BIMu pro systém DPS.....	21
3.4. Parametrické dynamické prvky systému DPS.....	22
3.4.1. Kotvy systému DPS .....	25
3.4.2. Nosník systému DPS .....	28
3.4.3. Sloup systému DPS .....	36
3.4.4. Stěna nosná .....	49
3.4.5. Stěna ztužující systému DPS.....	50
3.4.6. Stropní nosník Spiroll .....	54
3.4.7. Základový práh systému DPS .....	57
4. Dynamo .....	59
4.1. Představení Dynama.....	59
4.2. Nody, custom nody a wiry .....	59
4.3. Proces propojování, typy vstupů, výstupů, návaznosti .....	59
4.4. Tvorba dynamo skriptu pro potřeby systému DPS.....	60
4.5. Implementace skriptu do projektu a ukázka funkčnosti .....	74
5. Model z dynamických prvků systému DPS.....	77
5.1. Tvorba modelu z prvků DPS.....	77
5.2. Převod analytického modelu ve formát IFC .....	80
5.3. Statická analýza v softwaru Autodesk Robot .....	83
5.4. Použití klasifikačního systému a tvorba výkazu výměr a výrobků.....	89
5.4.1. Porovnání modelu č.1 a č.2 .....	91
5.5. Tvorba statických výkresů .....	94
6. Další využití modelu z dynamických prvků DPS .....	98



6.1. Příklad tvorby harmonogramu a animace výstavby .....	98
7. Závěr.....	105
8. Literatura.....	106
9. Seznam použitých softwarů .....	107
10. Přílohy.....	107
11. Seznam obrázků .....	108

# 1. Cíl práce

Cílem diplomové práce je vytvořit knihovnu dynamicky chovajících se prvků systému DPS, které budou odpovídat přesné kopii jejich reálnému představení. Tato knihovna prvků by měla sloužit k implementaci systému do projektů, které jsou vytvářeny v BIMu. Součástí práce, pro lepší uživatelskou manipulaci, by měl být vytvořen plugin do Revitu, v kterém se systém DPS parametrizuje. Tento plugin by poté měl sloužit jako pomocný nástroj pro vkládání prvků s přednastavenými parametry. Součástí práce by mělo být využití prvků na vzorovém modelu fiktivní budovy a následně tvorba dokumentace. Prvky jsou taktéž modelovány tak, aby měly přímou vazbu na analytické zobrazení a model byl tak kompatibilní se statickými softwarey. Výsledkem by měla být knihovna dynamických prvků DPS, skript pro implementaci těchto prvků, tvorba dokumentace a jednoduchá statická analýza.

## 1.1. Použité pojmy a zkratky

BIM – Building Information Model (Informační model stavby)

BEP – Bim Execution Plan (Plán Zpracování BIMu)

EIR – Employer's Information Requirments (Požadavky objednatele na BIM)

LOD – Level of Development

LOD – Level of Detail

LOI – Level of Information

DPS – Demontovatelný stavební systém

Integer – Číslo celé

Double – Desetinné číslo

String – Text

C# – programovací jazyk

## 2. Úvod do informačního modelování

### 2.1. BIM a implementace v rámci ČR a Evropy

BIM je zkratka pro Building Information Modeling nebo Building Information Model, v překladu informační model stavby nebo informační modelování stavby. Jedná se o tvorbu modelu, který slouží jako databáze dat pro zpracování dokumentace, tvorbu výkazů ale i ve fázi provozu budovy nebo její demolice. Důležité je ale právě slovní spojení databáze dat, jelikož se primárně jedná právě o tuto vlastnost, kterou BIM model obsahuje.

V ČR se k implementaci BIMu přistupuje poměrně pomalu a již byla dvakrát odložena. V rámci agentury ČAS ve spolupráci s Ministerstvem průmyslu a obchodu vzniká normalizace BIMu v rámci zakázek financovaných z veřejných zdrojů s rozpočtem nad určitou cenovou hodnotu. Tato implementace měla být do stavebnictví zavedena prvoplánově v roce 2019, následně to bylo posunuto na rok 2021 a poté zase na rok 2023.

### 2.2. Digitální dvojče jako databáze informací

BIM modelu se taktéž přezdívá digitální dvojče stavby. Ideologie tvorby modelu se může překloupat na proces stavby, tedy model budujeme od základů po stěny, stropy a střechu přesně jako reálnou stavbu. Výsledkem by měl být BIM model, tedy digitální dvojče, jako přesná kopie reálně provedené stavby. Ideologie tvorby informačního modelu není pouze pro fáze dokumentace, které známe ale může a funguje i jako zdroj pro správu objektu a na konci životnosti stavby i jako poklad pro bourací práce a demolici nebo demontáž stavby. V ČR je celkem běžné, že se BIM modely využívají nejen v projektové přípravě ale i při správě objektu až po její konec životnosti a šetří tak zbytečné náklady na procesy spjaté s procesy v životním cyklu budovy.

Co se týče podrobnosti, v jaké má být model vytvořen, ať už informační nebo grafické, ta se stanovuje takzvaným LOD, tedy level od development. Tato dokumentace obsahuje část obdobně nazvanou LOD a část LOI, tedy level of detail a level of information. Tyto hodnoty, které mají stupnici různou ale nejčastější 100, 200, 300, 400 a 500 určují právě v jaké podrobnosti, ať už jde o množství informací nebo grafické zobrazení má být model zpracován. Tato dokumentace, nebo její část je často obsažena v EIRu a následně v BEPu. EIR stanovuje požadavky na BIM model od investora na zhotovitele a BEP naopak stanovuje, jak bude BIM model zpracován zhotovitelem, tedy projektantem pro investora a dle požadavků stavby. Tyto dokumenty jsou nedílnou součástí výběru dodavatele dokumentace, pokud má být zpracována v BIMu a tvoří smluvní závazek mezi dodavatelem a investorem.

### 2.3. Revit, nástroj na tvorbu BIM modelu

V našem případě budeme tvořit BIM model v softwaru Autodesk Revit. Prvky v Revitu se modelují pomocí rodin anebo pomocí modelu na místě. Rodiny se dělí na systémové, uživatelské a komponenty na místě. Toto rozdělení je typické pro Revit a rodiny se takto dělí na základě jejich možností úpravy uživatelem. Tedy rodiny systémové jako například stěny, podlahy a střechy jsou editovatelné pouze v projektu. Rodiny uživatelské se modelují mimo projekt, jako samostatné soubory a následně se do projektu importují jako soubory rodiny s přívlastkem .rfa. Tyto rodiny zakládáme na šablonách a přiřazujeme jim kategorie, dle které se v Revitu zatřídí. Rodiny mají také dle kategorií a jazyku předinstalované šablony, na kterých se modelují. Tyto šablony se nedají vytvářet nebo upravovat a uživatel si tak musí vystačit s těmi dostupnými.

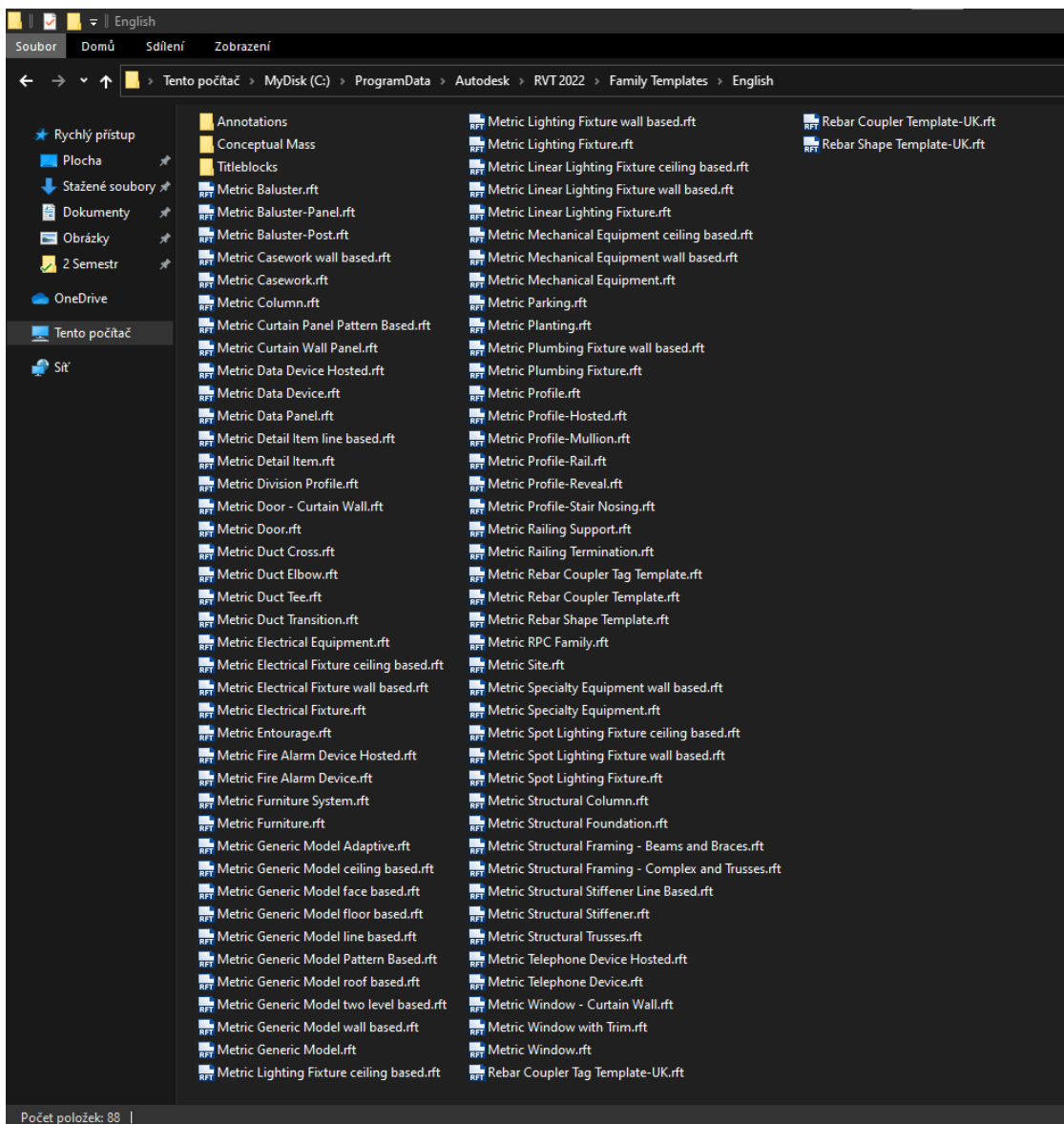
Jak jsem psal výše, rodiny uživatelské se modelují na určitých šablonách. Rodiny systémové jako stěny, schodiště a podobně bohužel šablony pro tvorbu rodin nemají a pokud se rozhodneme, například vytvořit rodinu stěny na šabloně obecný model a přiřadit mu následně kategorii stěny, tak to také nelze. Kategorie totiž také můžeme přiřazovat pouze u takových typů stavebních prvků, které jsou v Revitu brány jako rodiny uživatelské. Všechny tyto aspekty si představíme v práci dále.

## 3. Systém DPS s využitím BIMu

### 3.1. Parametrické navrhování

Systémové rodiny jsou většinou základními stavební prvky a jsou upravitelné pouze v modelu. Úprava a hlavně parametrizace, jak grafická, tak informační, rodin v modelu je limitována samotným Softwarem. Tento fakt si vysvětlíme následně na stavebním prvku stěna, v textu dále.

Uživatelské rodiny jsme naopak oproti systémovým daleko více schopni parametrizovat. Tyto rodiny jsou vytvářeny jako samostatné soubory na šablonách, které jsou součástí instalace softwaru Revit v různých jazycích ale i různých jednotkách. Pokud budeme mluvit o Revit šabloně, často je myšlena šablona modelu, které se dají vytvářet uživatelem a následně i upravovat. Toto nejsou šablony rodin, které, jak jsem psal výše, upravitelné nejsou. Rodiny uživatelské mají vlastní modelovací prostor a dají se parametrizovat jak geometricky, tak informačně. Každá šablona rodiny má různě přednastavené prvky, jako šířka profilu, délka tažení, a tak je třeba mít předem vymyšlený postup tvorby těchto rodin. Typy šablon můžeme vidět na obrázku 1. Taktéž je třeba zmínit, že rodiny mohou být nejen rodiny stavebních prvků ale i rodiny koncepčních objemů, rohového razítka nebo popisků. V Revitu existují 4 typy popisků, jsou to popisky kategorie, popisek multikategorie, popisek materiálu a popisek místnosti. Pro nás bude důležitý právě popisek multikategorie, který si představíme dále v práci při značení stavebních prvků systému DPS.



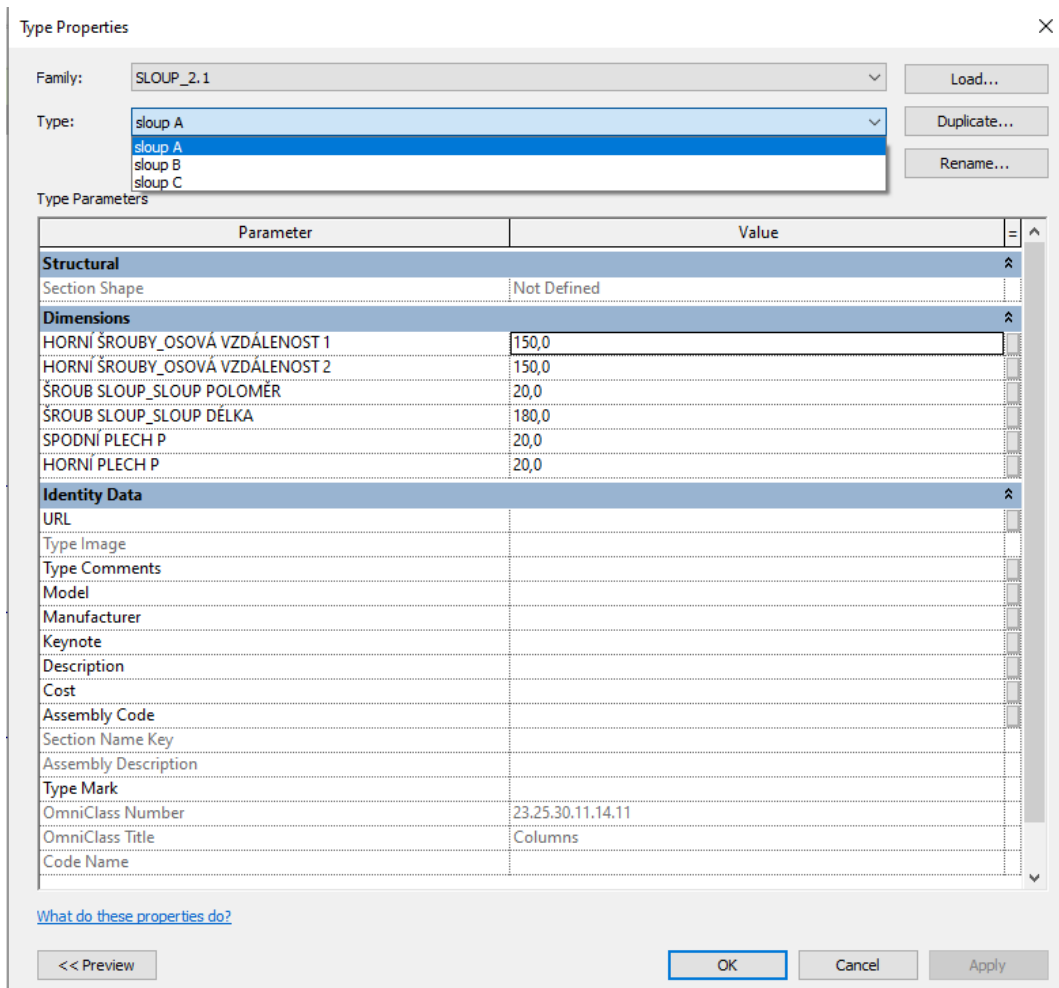
Obrázek 1 - Šablony rodin

Můžeme si všimnout, že je velké množství šablon pojmenováno Metric Generic Model neboli metrický obecný model. Tyto šablony mají často v sobě velice malé množství přednastavených parametrů, které mohou následné chování rodiny v modelu ovlivňovat. Tato šablona je dle mé dosažené praxe asi nejvíce využívána, neboť můžeme rodinu následně přiřadit jakékoliv kategorii, bez nutnosti zakládat rodinu na šabloně této kategorie a parametry si vytvořit dle požadavků prvku, který modelujeme.

Posledním typem rodin je komponenta na místě. Komponenty na místě jsou prvky modelované přímo v modelu a jejich kopírováním v modelu se duplikují rodiny se stejným názvem a rostoucí koncovkou 1 - n. Tuto možnost používáme pouze v případě tvorby prvku, který bude v celém projektu unikátní a nebude se duplikovat, tedy nám nebude v modelu

narůstat počet rodin a zvyšovat se tak jeho datový objem, který je u modelu důležitý, pro jeho plynulý běh.

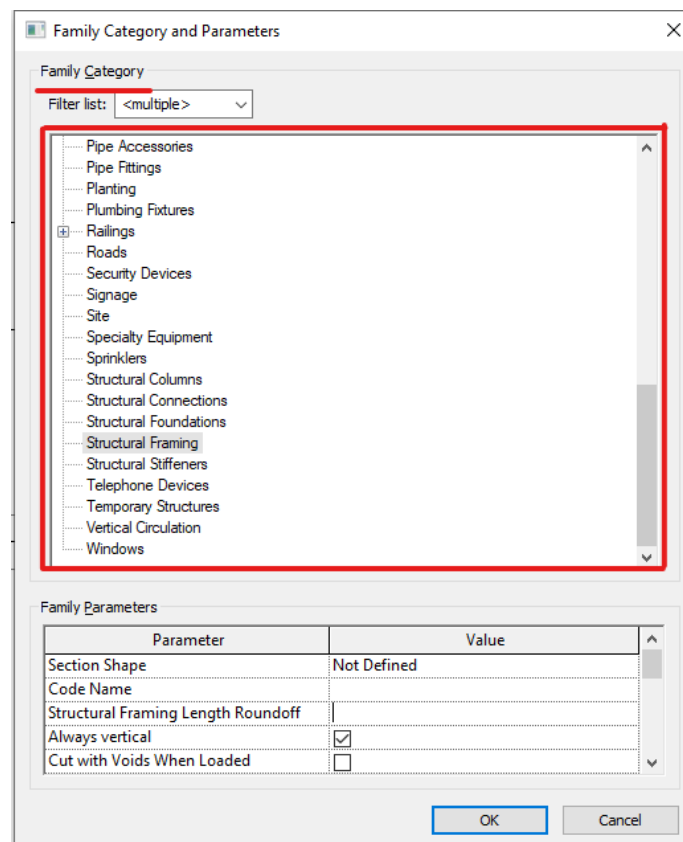
Co se týče rodin systémových a uživatelských je nutné zmínit jejich strukturní posloupnost. Máme rodinu, pod ní je typ a následuje instance. Instance je koncový prvek rodiny v modelu. Na základě této hierarchie se dělí i parametry, ke kterým se dostaneme následně. Do projektu načteme rodinu, která má 1 a více typů. Tyto typy můžeme duplikovat, přejmenovat a měnit jejich typové parametry. Dle jednotlivých typů následně modelujeme instance, těmito instancím poté můžeme nezávisle na sobě měnit instanční parametry. Tedy když změním parametr instanční, nezmění se u každé stejné rodiny v modelu, pokud ale změním parametr typový a v modelu máme těchto prvků více, změní se tento typový prvek u všech instancí stejné rodiny. Na obrázku 2 můžeme vidět pracovní verzi rodiny sloupu s třemi různými typy.



Obrázek 2 - Typové parametry rodin

Prvky, které se v systému DPS objevují a je tak nezbytné je parametrizovat jsou sloupy, stěny nosné, stěny ztužující, stropní nosníky, schodiště, základové konstrukce, průvlaky a kotvy, které jsou součástí systému.

Revit jako software na tvorbu BIMu má velké množství zabudovaných parametrů, které definují vlastnosti prvků. Jedním z nich je například účel prvků z hlediska statiky, tedy jestli se jedná o nosný nebo nenosný prvek. V našem případě se to týká všech výše popsaných konstrukčních dílců, které jsme schopni dělit na konstrukční a architektonické. Rozdíl ze stavařského hlediska je naprosto jasný, avšak v Revitu to znamená, že pokud vymodelujeme prvek jako konstrukční, tak bude součástí prvku i model analytický. Toto u prvku architektonického nenajdeme. Stejně to funguje i se šablonami. Pokud založím rodinu sloupu na šabloně metrický obecný model, jsme schopni vymodelovat prvek nosníku, přiřadit mu parametry a načíst do projektu jako nosník ale prvek nebude obsahovat model analytický, tedy prvek, který je pro převod výpočetního modelu pro analýzu konstrukce naprosto majoritní. Co bychom měli při tvorbě rodin ještě zmínit, je jejich zatřídění do kategorie. Rodinu nosníku jsme schopni založit na více různých šablonách a následně mu přiřadit kategorii viz obrázek 3. Prvky se dají zatřídit do všech kategorií uživatelských rodin, avšak nedají se zatřídit do kategorií rodin systémových. Není tedy možné založit rodinu na šabloně například obecný model a přiřadit mu kategorii stěna, strop nebo střecha.



Obrázek 3 - Zatřídění rodin do kategorií

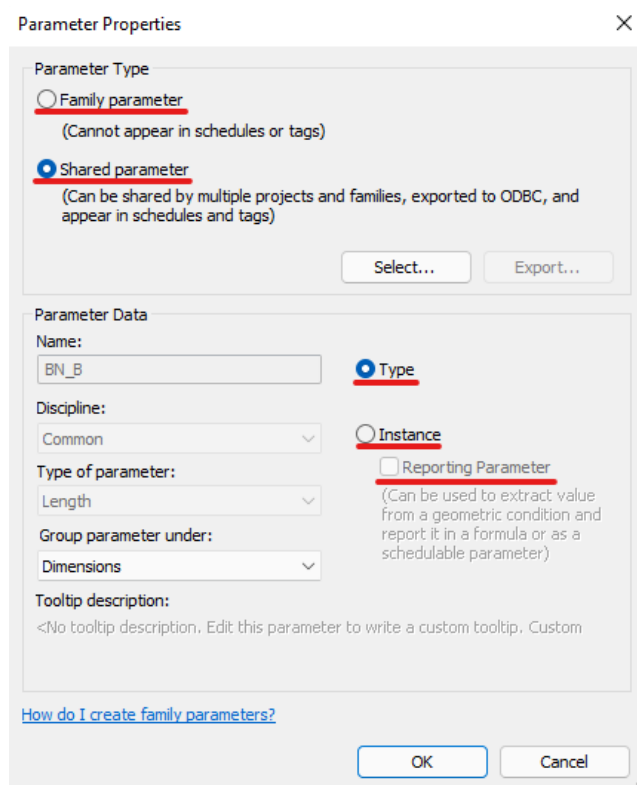


Stejně jako rodiny se dělí i parametry, které můžeme v Revitu využívat. Rozdělení je nejlépe vysvětleno na schématu viz. obrázek 4.

Parameter Type	Description	Examples
Project Parameters	Project parameters are specific to a single project file. They are added to elements by assigning them to multiple categories of elements, sheets, or views. Information stored in project parameters cannot be shared with other projects. Project parameters are used for scheduling, sorting, and filtering in a project.	A project parameter can be used to categorize views within a project.
Family Parameters	Family parameters control variable values of the family, such as dimensions or materials. They are specific to the family. A family parameter can also be used to control a parameter in a nested family by associating the parameter in the host family to the parameter in the nested family.	Family parameters such as Width and Height may be used in a Door family to control the dimensions of the different door types.
Shared Parameters	Shared parameters are parameter definitions that can be used in multiple families or projects. After you add a shared parameter definition to a family or project, you can use it as a family or project parameter. Because the definition of a shared parameter is stored in a separate file (not in the project or family), it is protected from change. For this reason, shared parameters can be tagged and scheduled.	If a parameter in a family or project needs to be scheduled or tagged, that parameter must be shared and loaded in both the project (or element family) and the tag family. Shared parameters can be used when elements in 2 different families are scheduled together. For example, if you need to create 2 different Isolated Foundation families, and need the Thickness parameter of both families scheduled in the same column, Thickness needs to be a shared parameter that is loaded in both Isolated Foundation families.
Global Parameters	Global parameters are specific to a single project file, but are not assigned to categories. Global parameters can be simple values, values derived from equations, or values taken from the model using other global parameters. Use global parameters to drive and report values.	A global parameter can assign the same value to multiple dimensions. You can also set the position of one element by the size of another element. For example, beams can be driven to consistently offset from the floor they support. If the floor design changes, beams will respond accordingly.

Obrázek 4 – Typy parametrů softwaru Revit

Parametry výše zmíněné se ještě dále dělí na instanční a typové. Instanční poté ještě na ty, které můžeme vykazovat a ty které vykazovat nemůžeme. Pokud avšak nastavíme instanční parametr projektový, jako parametr, který chceme vykazovat, není možné jej následně v projektu upravovat. Detail tvorby parametrů můžeme vidět na obrázku 5.

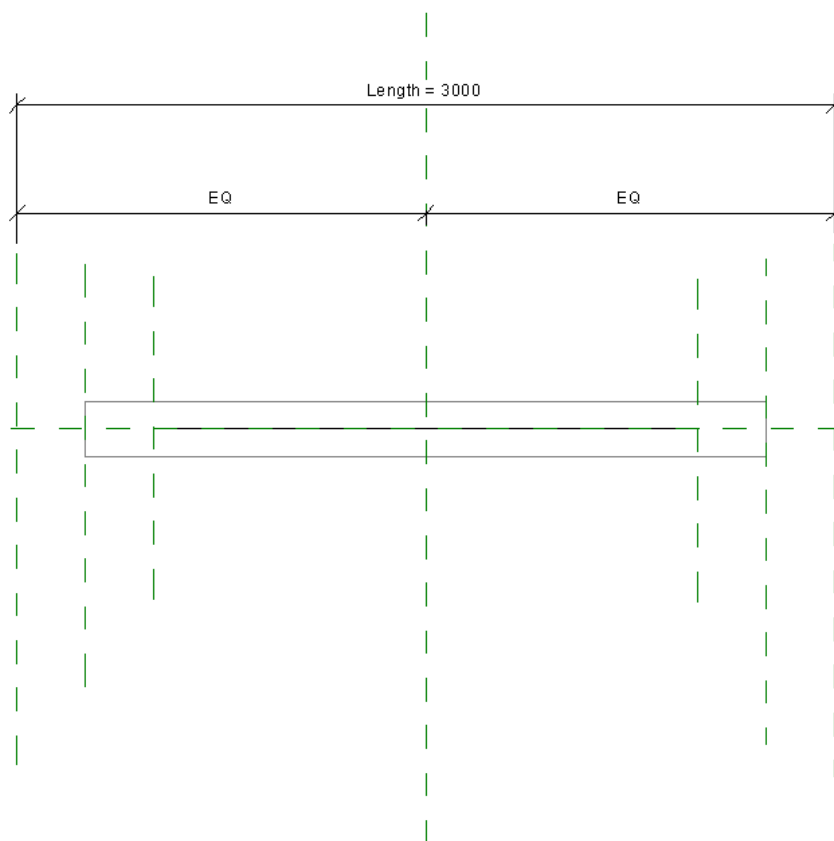


Obrázek 5 - Tvorba nového parametru

### 3.2. Obecné modelování prvků systému DPS

Prvotní verze nosníku byla modelována na dvou různých šablonách. Na šabloně obecný model liniový a konstrukční rámová konstrukce. Nosník vytvořený na šabloně obecný model liniový se choval uživatelsky přívětivěji oproti druhé variantě, nicméně jsme se v tomto případě připravili o analytický model prvku. Jelikož je to pro nás ale jedna z důležitých vlastností modelu, byla tato metoda i pro další konstrukční prvky zavržena.

Druhou a správnou možností bylo založení rodiny na šabloně konstrukční rámová konstrukce. Při prvním otevření šablony uvidíte nosník jako hmotu v půdoryse a několik referenčních rovin viz obrázek 6.

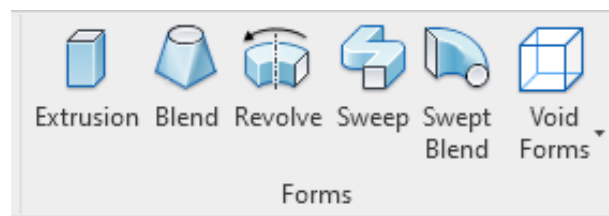


Obrázek 6 - Příklad modelovacího prostředí rodiny nosníku

Referenční roviny a čáry se při modelování rodin používají k uchycení geometrie hmoty. Obecně se dá říct, že se parametrizují referenční roviny a čáry a k těm se následně uchycují a zamykají plochy a hrany hmoty. Referenční roviny si můžeme pojmenovat a určit jejich charakter. Co tím myslím je to, že se dá rovině nastavit parametr, zda je reference slabá, silná, levá, pravá, horní, dolní nebo určuje počátek. Toto rozdělení má vliv na chování

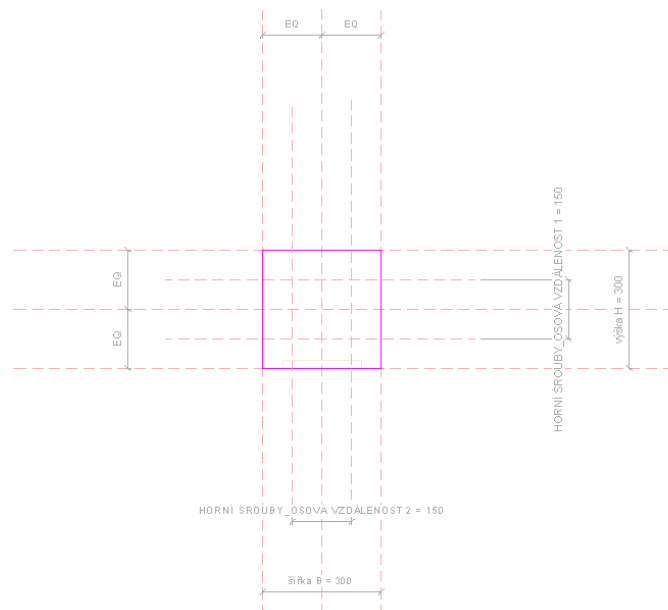
rodiny jako například, pokud určíme rovinu jako slabou, bude to mít v modelu vliv na kótování ve smyslu uchycení kóty k hraně hmoty, která je k rovině přichycena. Tedy nejdříve se bude kóta přichycovat k hranám s referenční rovinou silnou nebo levou či pravou a následně k rovině slabé. Zároveň můžeme nastavit, zda rovina určuje počátek. Pokud vezmeme dvě referenční roviny na sebe kolmé a určíme je jako počátek, průsečík těchto rovin bude určovat těžiště prvku a jeho zarovnání k ose procházející tímto bodem.

Tvorba hmoty v Revitu funguje na několika principech. Hmota samotná se dělí na solid a void. Solid je plná hmota, které můžeme přiřadit materiál. Void je hmota prázdná neboli dutý tvar, která slouží k ořezávání solid formy. Formy se přichycují k referenčním rovinám nebo čarám, ke kterým se poté zamykají, jak jsme zmiňovali již výše. Způsobů modelování máme více viz obrázek 7. Jednoduše popsáno na obrázku zleva doprava je to vysunutí, přechod, rotace, tažení a přechod tažením.



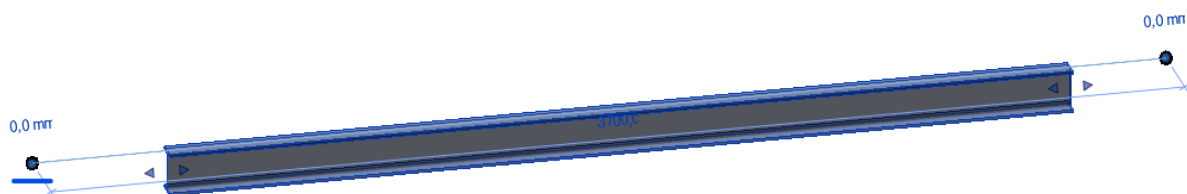
Obrázek 7 - Modelovací nástroje

Parametrizace těchto prvků probíhá, jak píšou výše pomocí referenčních rovin. Na obrázku 8 můžeme vidět růžovými čarami definovaný tvar hmoty vysunutím. Jelikož jsme v režimu úprav vše ostatní má potlačené barvy. Referenční roviny vytváříme mimo režim úprav hmoty, aby byly v rodině lépe dohledatelné. Po dokončení tvaru prvku protáhneme hrany hmoty k referenčním čarám a uzamkneme pomocí malého znaku zámku, který se nám objeví. Takto zajistíme, že hmota bude měnit tvar v závislosti na pozici roviny. Následně parametrizujeme roviny vytvořením kóty a přiřazením parametru právě této kóty.



Obrázek 8 - Příklad použití referenčních rovin

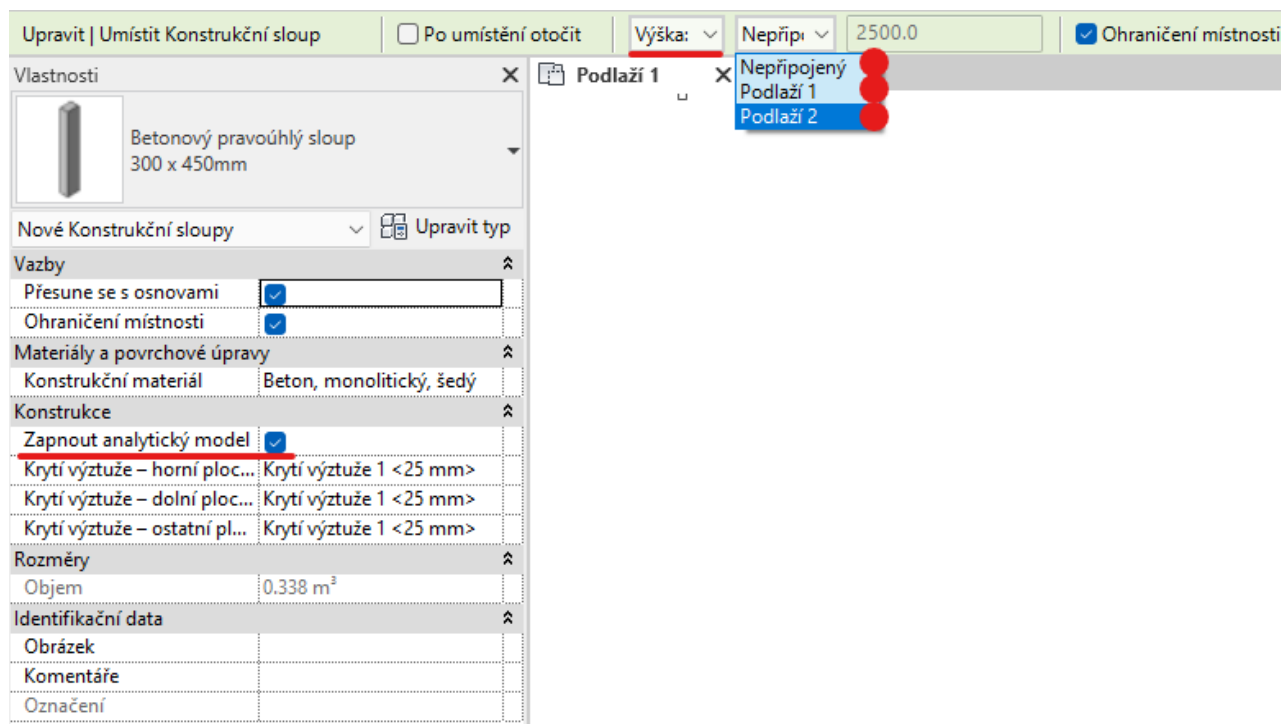
Chování nosníku v modelu je taktéž specifické a musí být při tvorbě nové rodiny bráno v potaz. Na obrázku 9 můžeme vidět Autodesk rodinu ocelového nosníku. Červeně jsou podtržené šipky, které upravují grafický model a modře kolečko, které upravuje model analytický. Jak můžete vidět, tyto dva parametry jsou nezávislé a vytváří u nosníku dvojici parametrů délky nazvané délka a délka řezu. Parametr délka značí délku modelu analytického délka řezu délku modelu grafického. Při vykazování délky nosníků je tedy důležité znát mezi těmito parametry rozdíl, aby nedošlo k navýšení délky celkové, a tak i špatnému nacenění položky.



Obrázek 9 - Typický model nosníku

Rodina sloupu taktéž jako nosník, funguje na určitém systému, podle kterého se modeluje. Na rozdíl od nosníku se totiž sloupu nenastavuje délka ale výškové rozmezí, většinou od podlaží k podlaží, popřípadě může být modelováno jako nepřipojeno a nastavena délka prodloužení počátečního a koncového. Stěny poté fungují obdobně s tím

rozdílem, že sloupy se modelují jednoduchým bodovým osazením na místo a stěny tažením v půdorysu, jelikož se jedná o prvek lineární. Tyto parametry můžeme vidět na obrázku 10, zároveň se můžeme všimnout parametru konstrukce zapnout analytický model, který určuje, zda je prvek konstrukční nebo architektonický, jak jsme psali výše.



Obrázek 10 - Modelování sloupu

### 3.3. Využití BIMu pro systém DPS

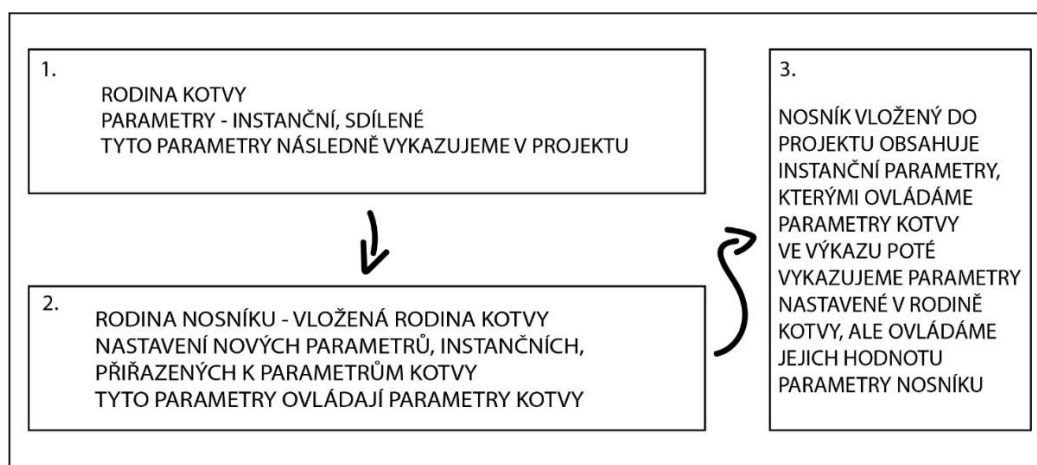
Systém DPS je demontovatelný, prefabrikovaný stavební systém, který obsahuje různé typy stavebních prvků. Jelikož se jedná o stavební systém demontovatelný, tak je v návrhu kladen vysoký nárok na spojovací prostředky. Pokud tuto složitost převedeme do BIMu, znamená to, že se jedná o prvky s poměrně velkým množstvím informací jak rozměrových, tak třídících, materiálových a podobně. LOD se tedy na těchto prvcích pohybuje na hodnotě zhruba 500 a LOI obdobně.

Jaké mohou být výhody využití BIMu pro DPS? No to je celkem zřejmé, tedy jeden prvek systému DPS, si nese tolik informací, které můžeme vykazovat, kombinovat apod., že může sloužit i jako zdroj pro výrobní dokumentaci s vazbou na každý prvek v modelu. Tím, že prvky obsahují i kotvy, které jsou plně parametrizované, je možné vytvářet výrobní dokumentaci i pro kotvy a její komponenty. Kotvy mají potom vazbu přímo na určitý prvek systému skrze klasifikační systém zavedený v našem typovém modelu budovy, kterou vysvětlíme.

### 3.4. Parametrické dynamické prvky systému DPS

Stavební systém DPS je poměrně komplexní systém, který nabízí velké množství variant a prvků. Vzhledem k možnosti demontáže systému jsou kladeny vysoké požadavky na kotvy spojující jednotlivé stavební prvky. Obecně jsou prvky modelovány jako rodina jedna a kotvy jako rodina druhá. Následně je rodina kotvy nahrána do rodiny nosného prvku. Všechny parametry kotvy jsou instanční sdílené, abychom je mohli vykazovat. Po nahrání rodiny kotvy, do rodiny např. nosníku, jsou vytvořeny duplicitní parametry v rodině nosníku, tedy stejné kategorie, zaražení, názvu ale s tím rozdílem, že jsou parametry instanční projektové, tedy nejsou sdílené. Tyto duplicitní parametry jsou poté přiřazeny referenčním parametrům kotvy. V modelu poté prvky fungují tak, že vykazujeme parametry kotvy, ale ovládáme je parametry v nosníku jako je naznačeno ve schématu viz obrázek 11. Postup přiřazování je pak znázorněn na obrázku 12.

Parametry kotev mohou být instanční i typové, avšak pokud jsou typové, v modelu se poté kotvy chovají jako jeden typ a nelze vytvořit typ nový. To znamená, že mohou mít libovolný počet nosníků, ale všechny budou mít stejnou kotvu, která bude měnit své parametry napříč celým projektem, a to není přijatelné. Abych tento jev obešel, musím parametry kotvy vytvořit dvakrát, tedy v rodině kotvy a následně v rodině nosníku. Tyto parametry musí být instanční je pak možné je v modelu na určitém nosníku měnit bez vazby na ostatní nosníky. Dalším důvodem je, aby nedocházelo k nadměrnému kopírování typů jedné rodiny, ale aby tak docházelo pouze v případě změny rozměrů nosníku nebo jeho jiných typových parametrů.

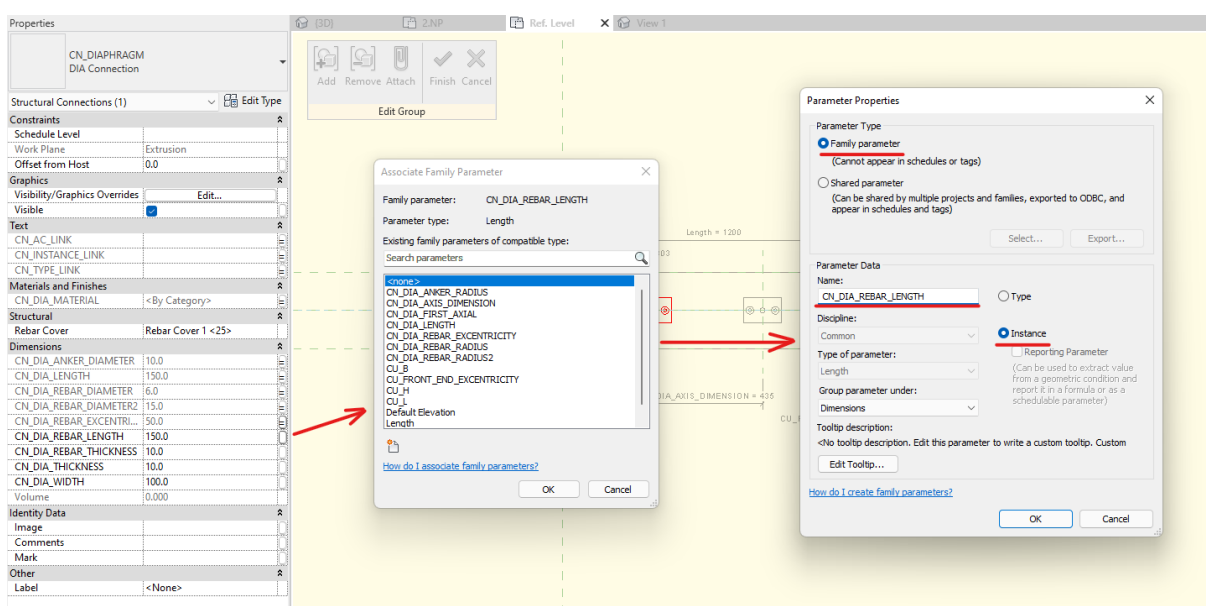


Obrázek 11 - Schéma parametrizace kotev

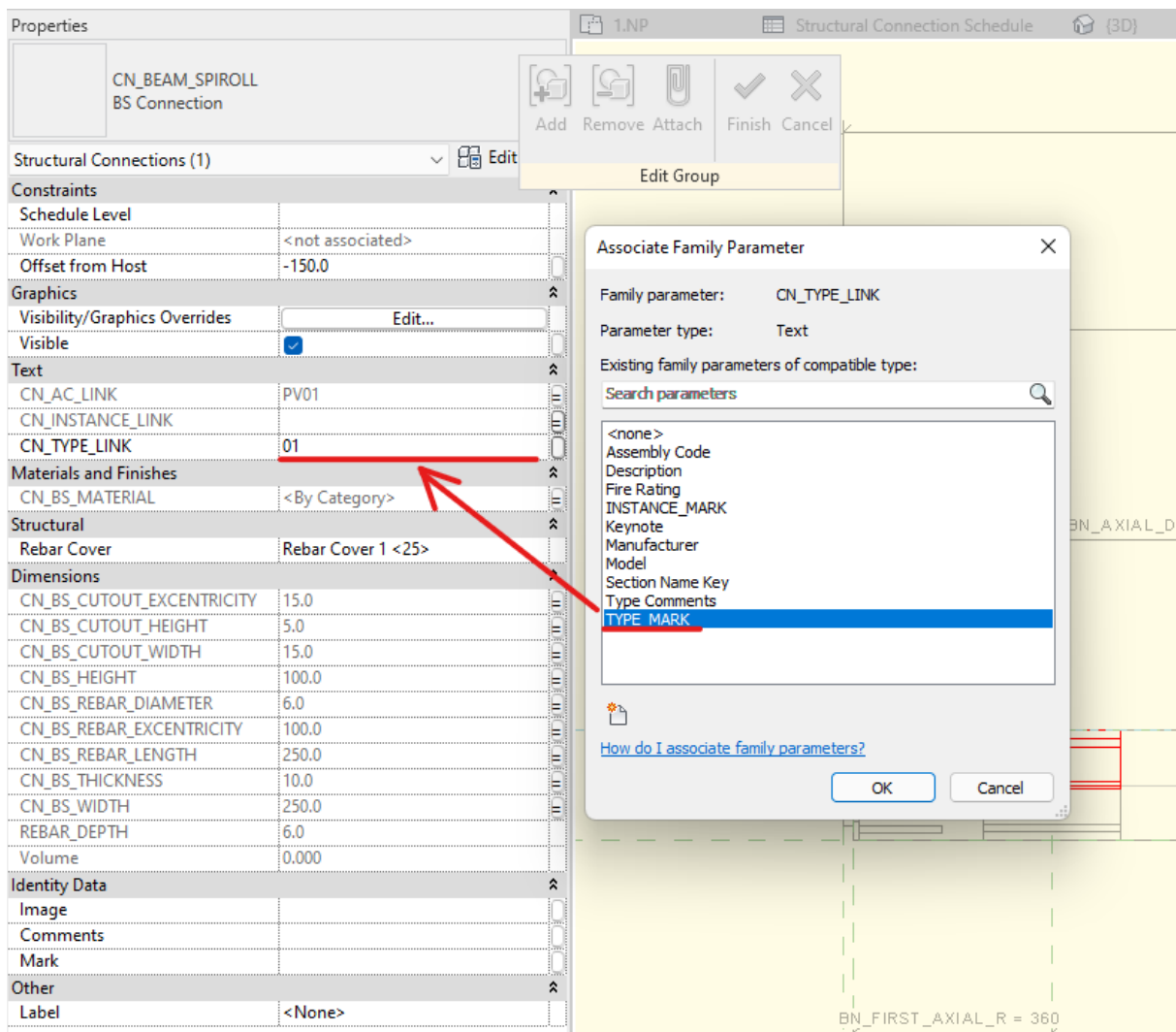
Tento postup nám dovoluje kontrolovat parametry kotvy na každém nosníku zvlášť bez ohledu na další, stejné instance v modelu. Původně byly parametry kotvy nastaveny v její rodině jako typové, aby každá instance, která bude mít rozdílné parametry kotvy, byla nová

typ. Nicméně pokud jsme vytvořili nový typ nosníku, nevytvořil se nový typ kotvy a kotvy se měnili napříč projektem i mezi různými typy. Tedy nebylo možné vytvořit více typů nosníku s jinými rozměry kotvy jedné rodiny.

BIM je víceméně o informacích, které můžeme prvkům přiřadit a jelikož je systém, jak už jsem psal výše celkově komplexní, je třeba v něm nastavit i určitý třídící řád. Nosníky a jiné prvky jsou tříděny dle kódu sestavy, typového označení a označení instančního. Abychom byli schopni detekovat pozici kotev, vůči prvku, byli kotvám přiřazeny parametry, které určují vazbu k tomu určitému prvku. Tyto parametry jsou vytvořeny v rodině kotvy a následně jim jsou v rodině nosného prvku přiřazeny parametry kód sestavy, typové označení a instanční označení viz obrázek 13.



Obrázek 12 - Parametrizace kotev v rodině nosného prvku



Obrázek 13 - Vazba kotev na značení nosných prvků systému DPS

Jednotlivé prvky a jejich detaily jsou popsány níže. Jelikož jsem pracoval s tím, co nám Revit nabízí, nebylo možné parametrizovat úplně všechny prvky systému DPS, jako jsou právě rodiny systémové tedy stěny, schodiště nebo podlahy. Tyto prvky bychom samozřejmě vždy mohli vytvořit jako rodiny, zatřídit je do určité kategorie, i když Revit vám nedovolí, zatřídit rodinu do kategorie stěny, a následně je modelovat jako stěnu nebo schodiště. Nicméně problém je v tom, že bychom zase přišli o analytický model, který může ve výsledku zjednodušit práci dalším profesím. Tato problematika a její řešení je popsána detailněji dále v práci.



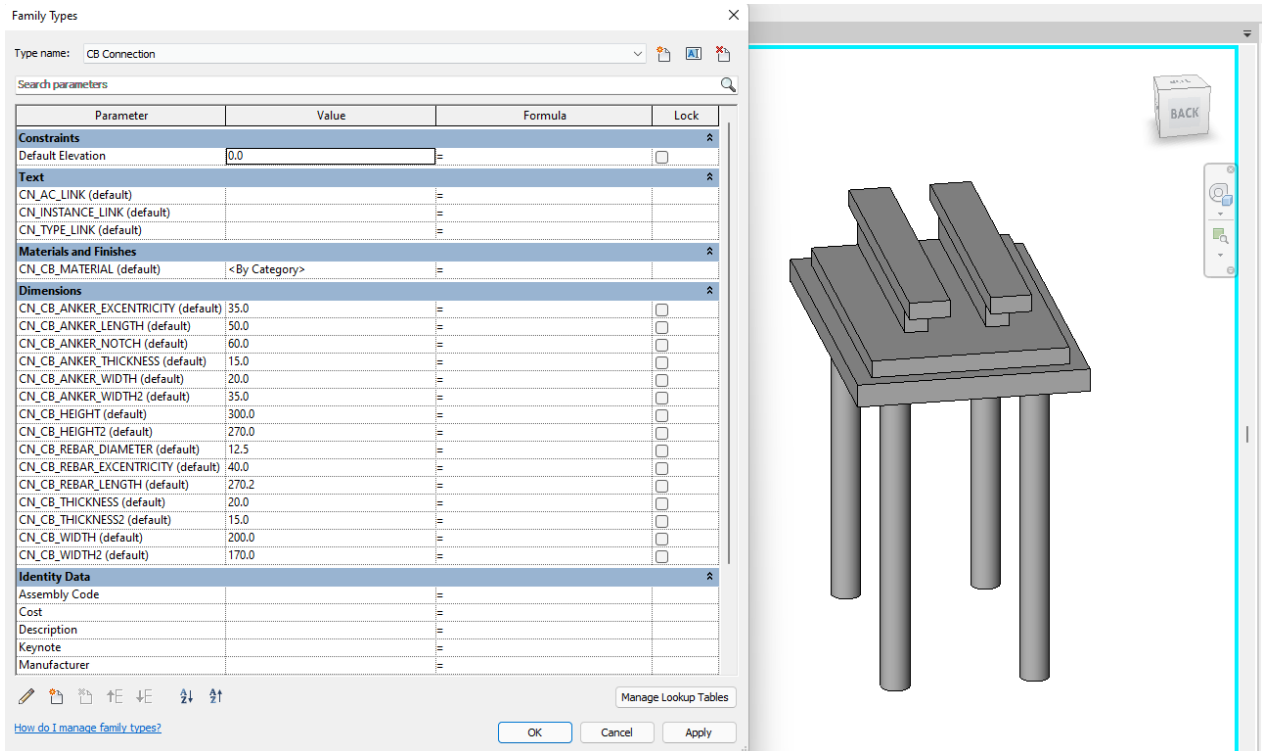
### 3.4.1. Kotvy systému DPS

Kotvy systému DPS jsou všechny modelovány na šabloně metrický obecný model a je jim přiřazena kategorie konstrukční přípoj. Kotvy jsou následně vloženy do rodiny nosníku, sloupu, ztužující stěny nebo základového prahu. Pokud je rodina vložena do rodiny jiné, říká se jí rodina vnořená. Kotvy mají přiřazené instanční, sdílené parametry, které ovládají jejich rozměry, označení a materiál. Pokud jsou kotvy vloženy do jiné rodiny a tato rodina následně do modelu, nelze ovládat parametry vnořené rodiny. Tedy abychom byli schopni ovládat parametry vnořených rodin, našich kotev, musí jim být přiřazen parametr i v rodině nosníku a podobně. O tomto procesu píšu dále v práci.

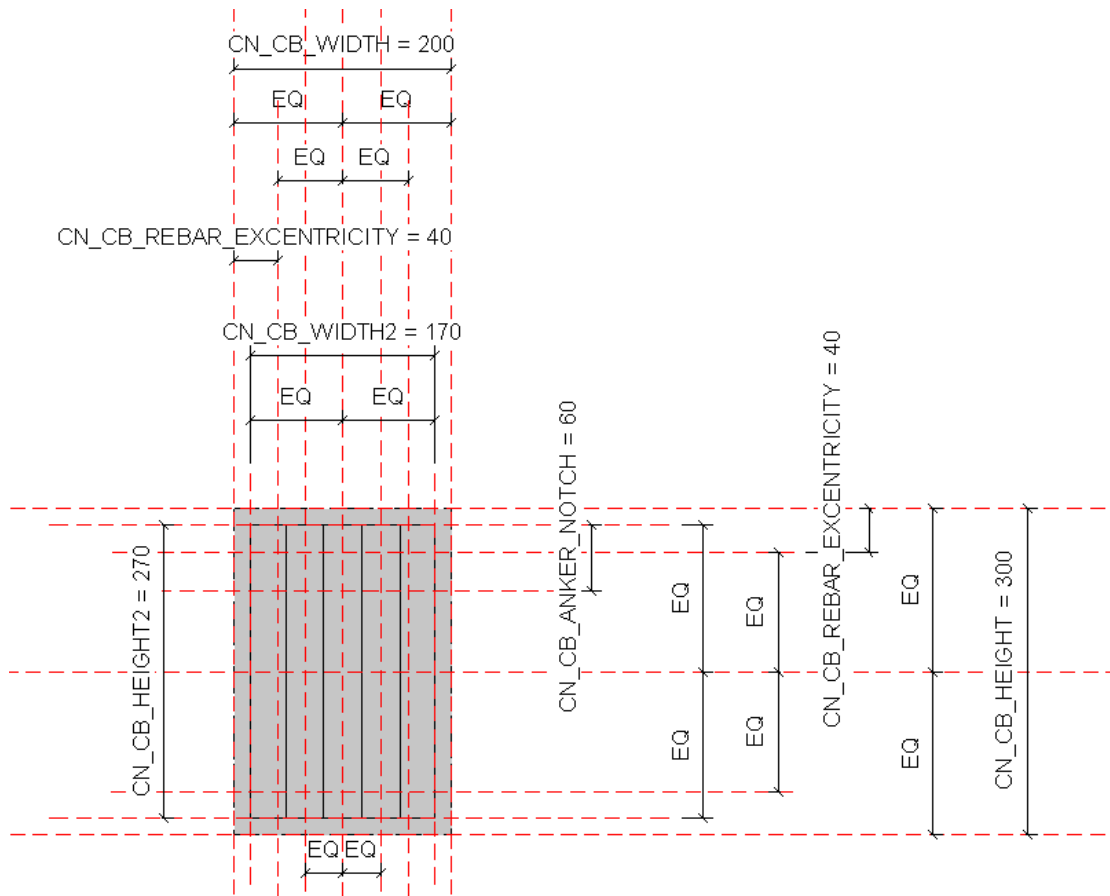
Každá kotva je modelována tak, jak by měla v praxi vypadat a být vyrobena. Tedy parametry kotvy můžou být výchozím výstupem pro jejich výrobu. Jelikož jsou ale kotvy poměrně složité, pouhé názvy parametrů, které kotvy nesou, nemůžou být jediným zdrojem informací. Byl tak vytvořen katalog prvků, které jsou v rámci diplomové práce vymodelované. Tento katalog prvků obsahuje všechny nosné prvky i kotvy se schémata a popsány parametry. Tento katalog je součástí diplomové práce jako příloha.

Kotvy jsou modelovány tak, jak by měly vypadat, nicméně jsou případy, kde vzhled kotvy není napřímo jasný a zároveň není přímo možné kotvu parametrizovat tak, abych pokryl všechny možnosti. Je tedy možné, že některé kotvy nejsou kompletní, například u kotvy sloup sloup, která může být čtvercová, kulatá, šestihranná může být několik verzí s pozicí spojovací výztuže s různým rozložením, počtem a podobně. Toto bohužel v rámci rodin nejsem schopen zachytit a jediné řešení je vytvořit více rodin v těchto variantách.

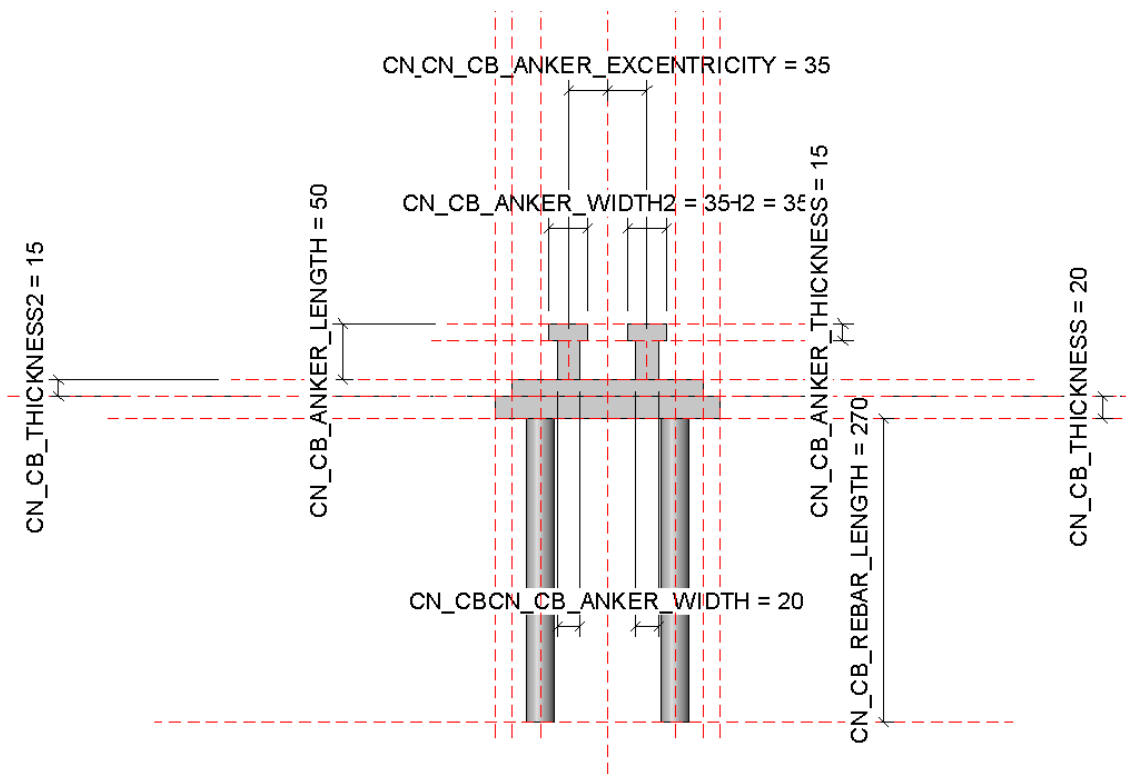
Dalším prvkem, který jsem si zjednodušil, byla kotva sloup nosník, která se skládá z kotev dvou. Toto je způsobeno právě výrobním procesem sloupu. Kotva se tak skládá z kotev dvou, které do sebe následně zapadnou. V našem případě jsem si ale prvek zjednodušil a kotvu vymodeloval jako prvek jeden. Tento rozsah je naznačen v katalogu prvku u kotvy sloup nosník. Příklad, jak jsou kotvy modelovány můžeme vidět na následujícím obrázku 14, 15 a 16, na kterých můžeme vidět 3 typy pohledů na kotvu sloup nosník.



Obrázek 14 - Příklad rodiny kotvy sloup nosník

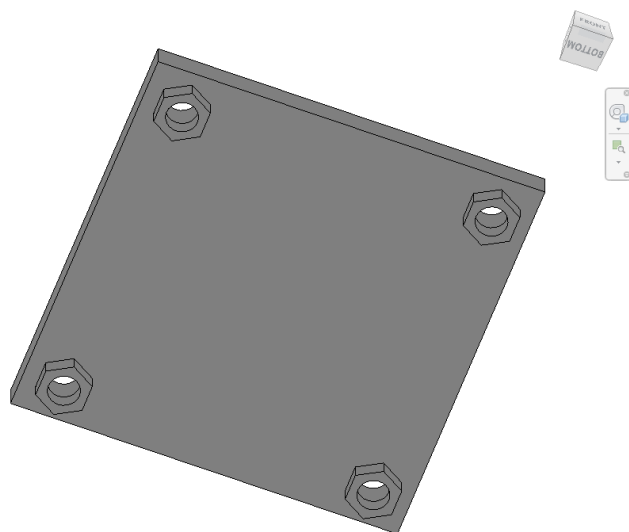


Obrázek 15 - Čelní pohled na kotvu sloup nosník



Obrázek 16 - Horní pohled na kotvu sloup nosník

Kotva sloup nosník je jedna z rodin složitějších, vzhledem k jejímu tvaru a počtu parametrů. Právě tato kotva je reálně rozdělena na dvě a je třeba její dokumentaci upravit. Kotvy a jejich parametry jsou značeny počátečním kódem ve tvaru CN poté například CB jako Column Beam a název parametru. Rodina kotvy je značena jako CN\_COLUMN\_BEAM a typ je značen jako CB Connection.

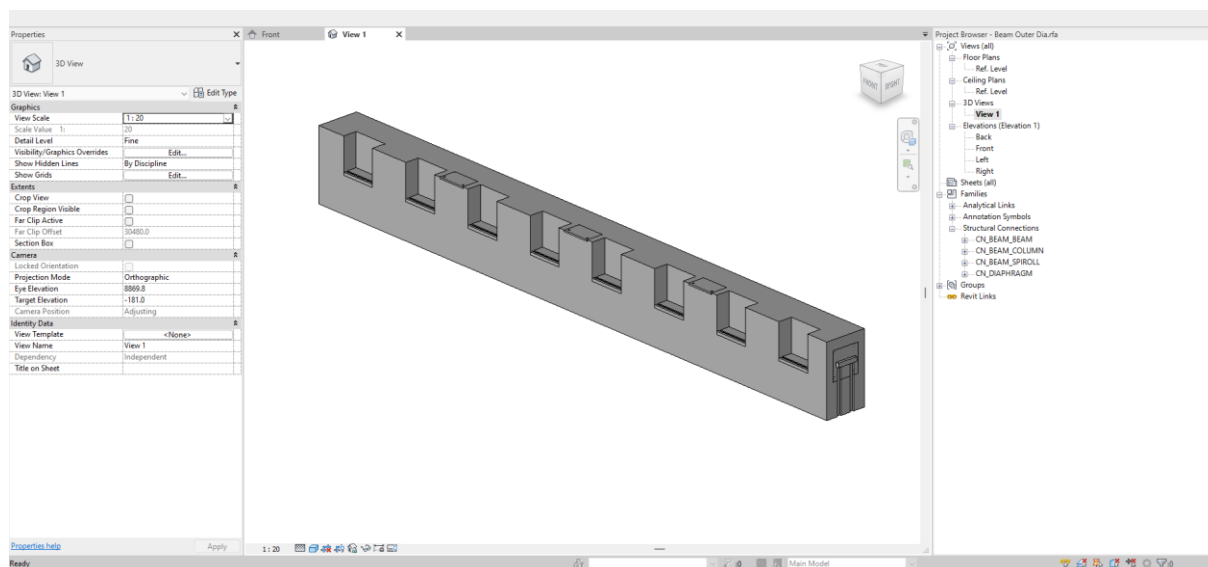


Obrázek 17 - Kotva sloup sloup horní

### 3.4.2. Nosník systému DPS

Nosník jako takový, v systému DPS tvoří asi nesložitější prvek na parametrizaci, současně se sloupem. Jelikož se jedná o prvek, který může obsahovat kotvy pro stropní nosníky, kotvy pro připoj ke sloupům anebo kotvy pro připoj ztužujících stěn. Zároveň může být nosník s kotvami pro stropní panely Spiroll z jedné strany nebo ze dvou stran anebo úplně bez kotev.

Nosník jako rodina je vytvořen na šabloně konstrukční rámová konstrukce. Zároveň je rozdělen na pět rodin, tedy krajní, středový a bez kotev pro stropní panely a následně ve stejném složení akorát s kotvami pro ztužující stěny. Toto rozdělení plyne z možností, které nabízí modelovací nástroje. Kromě základních věcí jako je příčný průřez a délka nosníku, se dá upravovat analytická délka, počet kotev, jejich osová vzdálenost a vzdálenost mezi první řadou kotev a krajem nosníku. Zároveň skrze nosník ovládáme parametry kotev. Parametry jsou názvem rozděleny dle počáteční zkratky, tedy pro nosník je to BN a pro kotvy je to CN jako connection. Jak už bylo výše zmíněno, kotvy jsou rodiny, založené na šabloně rodiny obecný model a je jim přiřazena kategorie konstrukční spoje. Na obrázku 17 můžeme vidět verzi nosníku s kotvami pro stropní panely z jedné strany. Můj model se oproti výše zmíněné rodině nosníku chová rozdílně, a to právě kvůli množství a složitosti parametrů přidaných do rodiny. Polohu a směr nosníku nastavujeme graficky, tažením analytického zobrazení a rozměry nastavujeme ručně vyplněním tabulky parametru prvku. Tedy spíše vyplněním hodnoty počátečního a koncového zkrácení. Analytická délka nosníku je tak rovná parametru Length a grafická délka nosníku parametru BN\_L. Na obrázku 18 máme rodinu nosníku ve 3D pohledu.



Obrázek 18 - 3D pohled rodiny nosníku

Nosník se chová dynamicky, tedy počet kotev reaguje na grafickou délku nosníku a osovou vzdálenost mezi kotvami. Pokud nosník protáhneme, počet kotev se navýší viz obrázek 21 a 22. Toto chování plyne z formule nastavené v rodině prvku viz obrázek 19, která počítá počet kotev z celkové délky grafické verze nosníku dělené osovou vzdáleností kotev. Tyto parametry se jmenují BB\_NUMBER\_OF\_ANCHORS a BS\_NUMBER\_OF\_ANCHORS.

Family Types

Type name: Default Beam

Search parameters

Parameter	Value	Formula	Lock
<b>Materials and Finishes</b>			
CN_BB_MATERIAL (default)	<By Category>	=	
CN_BC_MATERIAL (default)	<By Category>	=	
CN_BS_MATERIAL (default)	<By Category>	=	
Structural Material (default)	<By Category>	=	
<b>Dimensions</b>			
BN_B	200.0	=	<input type="checkbox"/>
BN_H	450.0	=	<input type="checkbox"/>
BN_FRONT_END_EXCENTRICITY (default)	20.0	=	<input type="checkbox"/>
BN_FIRST_AXIAL (default)	360.0	=	<input type="checkbox"/>
BN_AXIAL_DISTANCE (default)	540.0	=	<input type="checkbox"/>
BN_L (default)	4410.0	= Length - (2 * BN_FRONT_END_EXCENTRICITY)	<input type="checkbox"/>
CN_BB_DIAMETER (default)	6.0	=	<input type="checkbox"/>
CN_BB_EXCENTRICITY (default)	20.0	=	<input type="checkbox"/>
CN_BB_HEIGHT (default)	120.0	=	<input type="checkbox"/>
CN_BB_THICKNESS (default)	10.0	=	<input type="checkbox"/>
CN_BB_WIDTH (default)	200.0	=	<input type="checkbox"/>
CN_BC_DEPTH (default)	150.0	=	<input type="checkbox"/>
CN_BC_GROOVE_EXCENTRICITY (default)	40.0	=	<input type="checkbox"/>
CN_BC_GROOVE_HEIGHT (default)	120.0	=	<input type="checkbox"/>
CN_BC_GROOVE_WIDTH (default)	20.0	=	<input type="checkbox"/>
CN_BC_HEIGHT (default)	200.0	=	<input type="checkbox"/>
CN_BC_JOINT_HEIGHT (default)	30.0	=	<input type="checkbox"/>
CN_BC_JOINT_THICKNESS (default)	20.0	=	<input type="checkbox"/>
CN_BC_JOINT_WIDTH (default)	150.0	=	<input type="checkbox"/>
CN_BC_REBAR_DEPTH (default)	150.0	=	<input type="checkbox"/>
CN_BC_REBAR_EXCENTRICITY (default)	20.0	=	<input type="checkbox"/>
CN_BC_REBAR_RADIUS (default)	6.0	=	<input type="checkbox"/>
CN_BC_THICKNESS (default)	10.0	=	<input type="checkbox"/>
CN_BC_WIDTH (default)	150.0	=	<input type="checkbox"/>
CN_BS_DEPTH (default)	250.0	=	<input type="checkbox"/>
CN_BS_CUTOUT_EXCENTRICITY (default)	15.0	=	<input type="checkbox"/>
CN_BS_CUTOUT_HEIGHT (default)	5.0	=	<input type="checkbox"/>
CN_BS_CUTOUT_WIDTH (default)	15.0	=	<input type="checkbox"/>
CN_BS_HEIGHT (default)	100.0	=	<input type="checkbox"/>
CN_BS_REBAR_DIAMETER (default)	6.0	=	<input type="checkbox"/>
CN_BS_REBAR_EXCENTRICITY (default)	100.0	=	<input type="checkbox"/>
CN_BS_REBAR_LENGTH (default)	250.0	=	<input type="checkbox"/>
CN_BS_THICKNESS (default)	20.0	=	<input type="checkbox"/>
CN_BS_WIDTH (default)	250.0	=	<input type="checkbox"/>
Length (default)	4450.0	=	<input type="checkbox"/>
<b>Other</b>			
BB_NUMBER_OF_ANKERS (default)	3	= (BS_NUMBER_OF_ANKERS / 2) - 1	<input type="checkbox"/>
BS_NUMBER_OF_ANKERS (default)	8	= BN_L / BN_AXIAL_DISTANCE	<input type="checkbox"/>
<b>Identity Data</b>			

Manage Lookup Tables

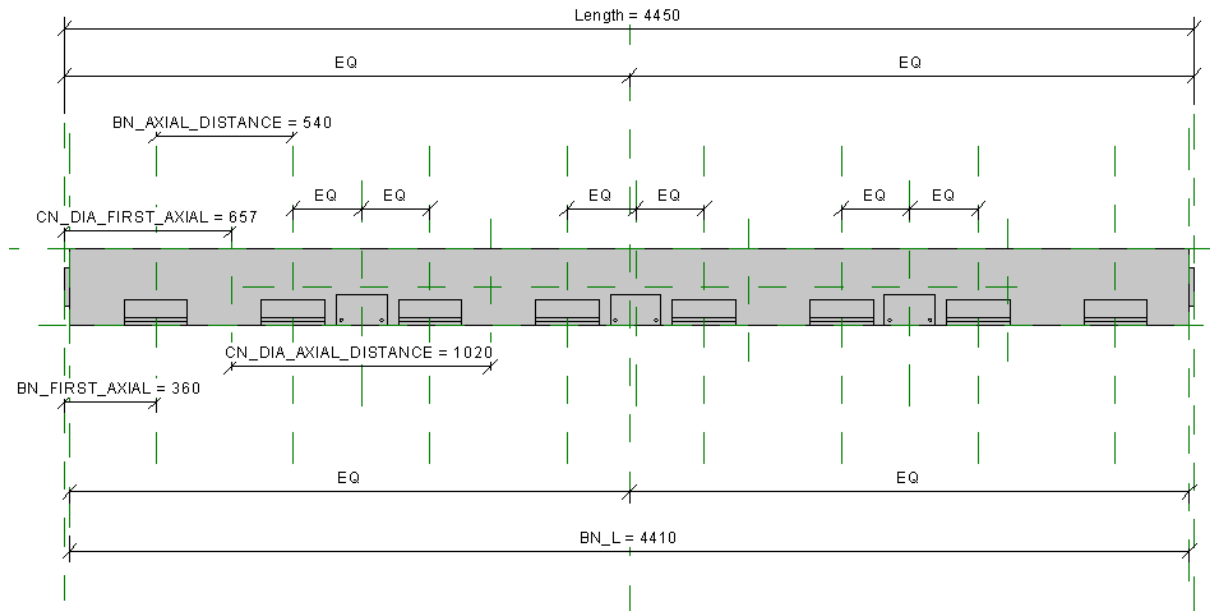
OK Cancel Apply

How do I manage family types?

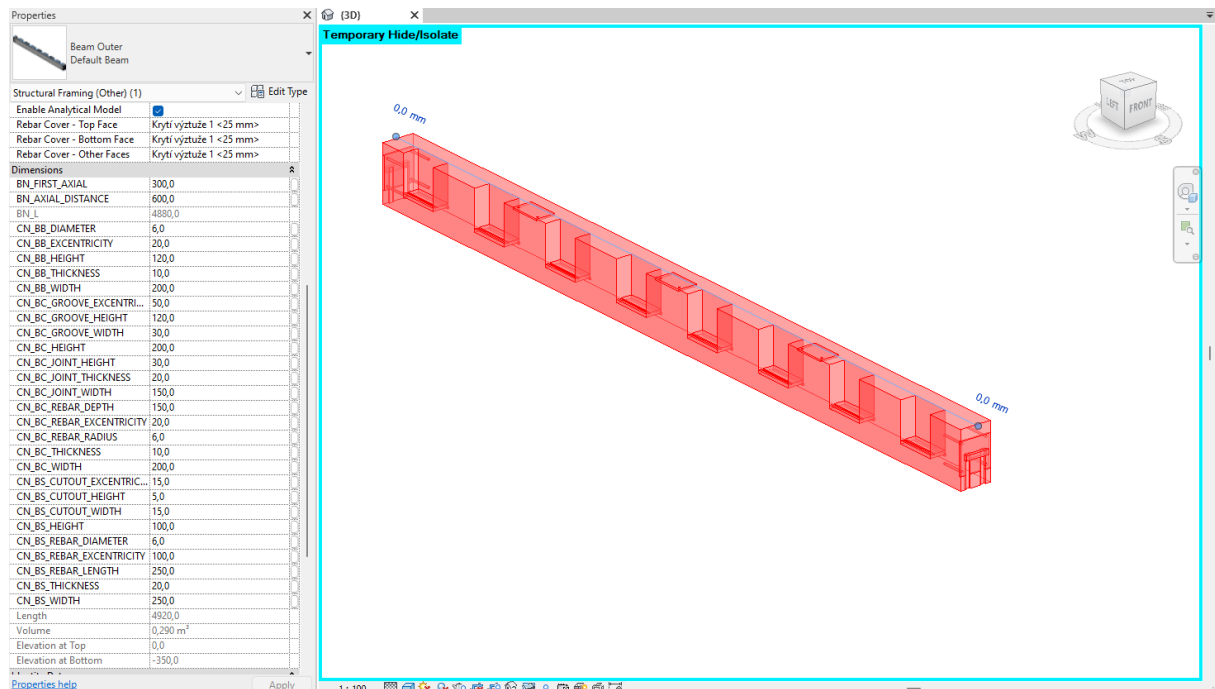
Obrázek 19 - Ukázka parametrů rodiny nosníku

Abychom si byly schopni představit, jak je nosník parametrizován, například přímo 20, kde jsou tyto parametry přiřazeny pod názvem BN\_L a BN\_AXIAL\_DISTANCE. Kotvy pro

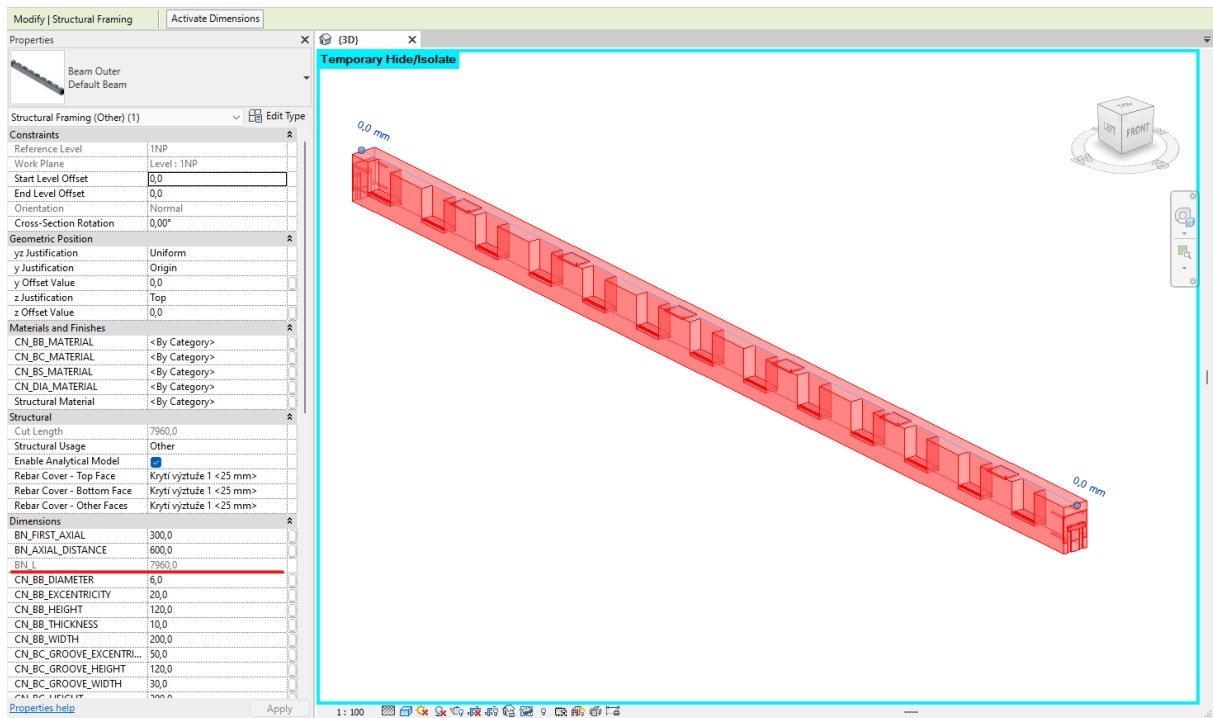
přípoj Spiroll nosník Spiroll, které na obrázku také vidíme, jsou parametrizovány podobně, až na jejich polohu, která je přímo ukotvena na liché osy mezi kotvami pro přípoj nosník Spiroll.



Obrázek 20 - Půdorysný pohled na rodinu nosníku

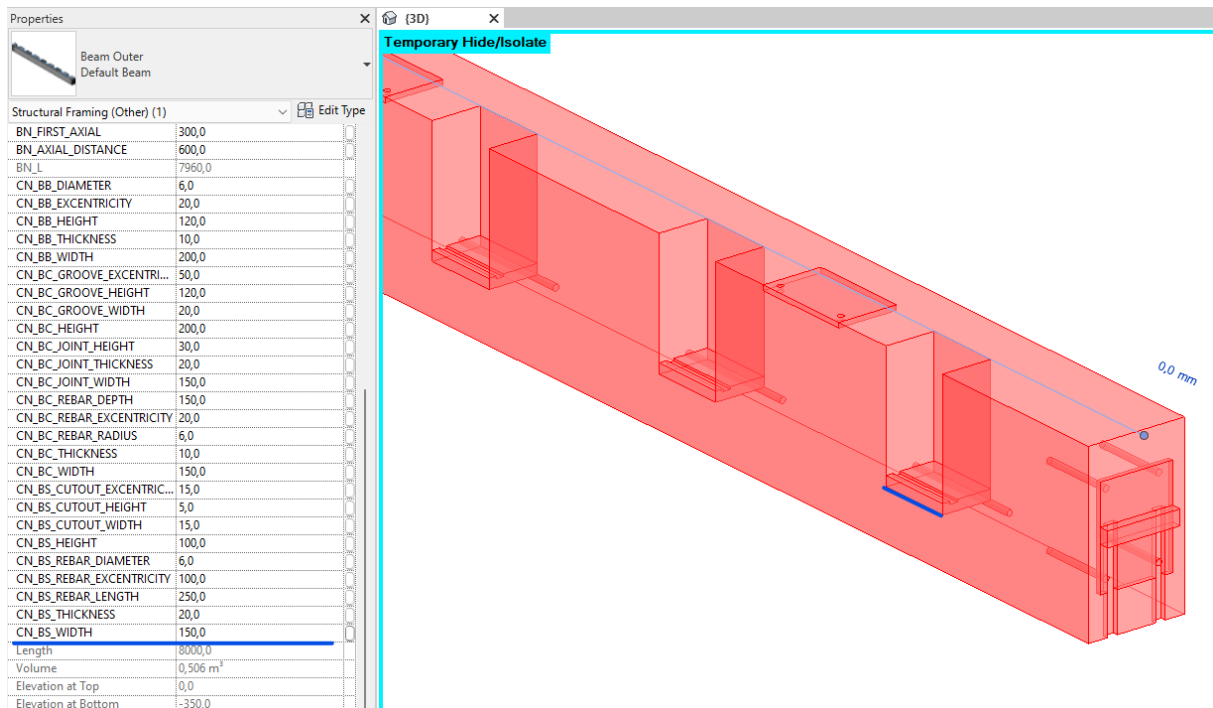


Obrázek 21 - Nosník systému DPS v modelu

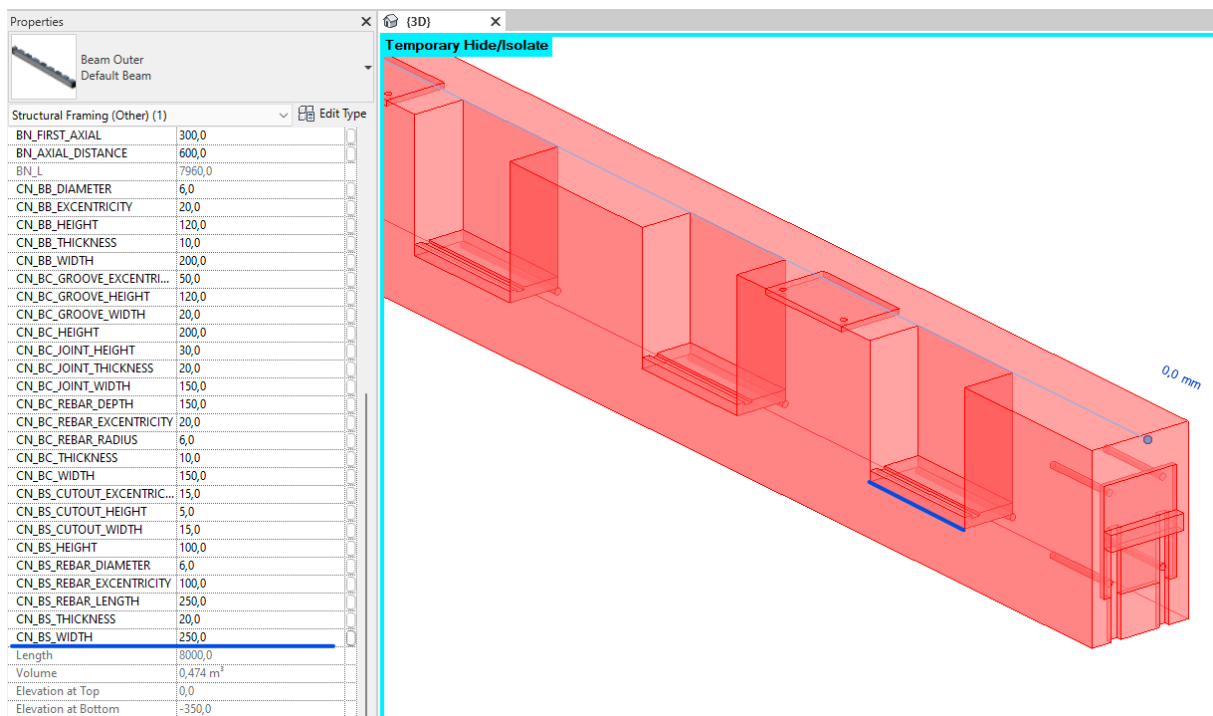


Obrázek 22 - Nosník systému DPS prodloužený

Kotvy pro stropní nosníky se chovají obdobně, pokud změníme jejich rozměr, změní se i výřez v nosníku v závislosti na velikosti kotvy viz obrázky 23 a 24.



Obrázek 23 - Nosník DPS před úpravou parametru kotvy



Obrázek 24 - Nosník DPS po úpravě parametru kotvy

Toto chování je zajištěno dutým tvarem, modelovaným tažením, který se přizpůsobuje tvaru kotvy a ořezává prvek nosníku. Všechny tyto atributy i s referenční rovinou, která určuje osu kotvy jsem následně rozkopíroval příkazem pole a přiřadil mu parametr počtu kotev, o kterém jsem psal výše.

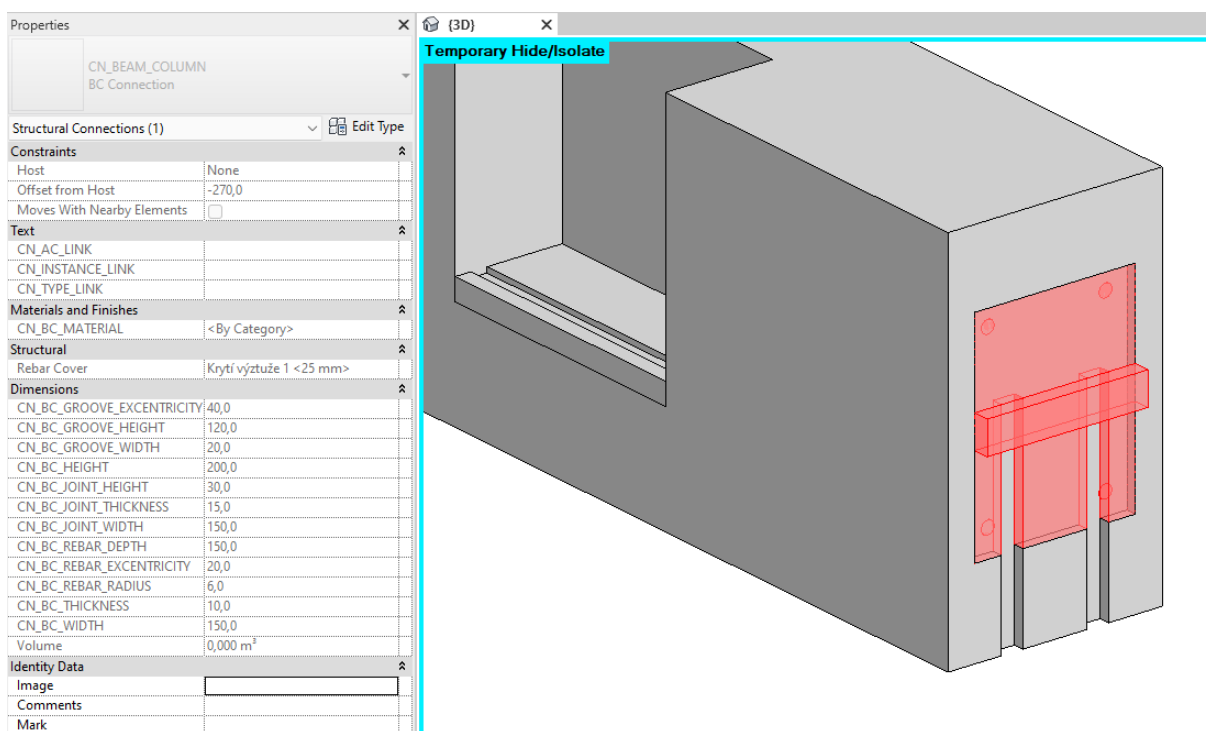
Kotvy, které jsou ve větším počtu na jednom stejném nosníku, se úpravou parametru změní právě na tomto celém jednom nosníku. Stejně kotvy, na nosníku stejného typu ale jiné instance ne. Pokud bychom chtěli změnit nosníku stejného typu a instance parametr napříč celým projektem, můžeme vybrat všechny tyto stejné instance stejného typu najednou a změnit jim hromadně instanční parametr, není tedy třeba to měnit pro každou instanci jednotlivě.

Proč mám prvky, jako například nosník rozdělený na vnitřní a vnější? To je dáno tím, že pokud nastavíme kotvám v nosníku parametr množení přes příkaz pole, automaticky se prvek vloží do skupiny. Skupina je v Revitu v podstatě soubor instančních prvků, skupiny se dají pojmenovat a najdeme je v projektovém prohlížeči. Skupiny se dají kopírovat napříč projektem, tedy vytváříme instance skupin a když jednu upravíme, propíše se úprava napříč všemi stejnými skupinami v modelu. Skupinám se však nedá upravovat parametr viditelnosti, avšak dá se upravovat viditelnost prvků ve skupině, takže bychom eventuelně nemuseli vytvářet víc rodin nosníků a měli pouze jednu, kde bychom upravovali viditelnost kotev vlevo a vpravo. Nicméně další problém vzniká s oříznutím místa pro kotvu, které je vytvořeno dutým tvarem. Dutému tvaru se bohužel nedá přiřadit parametr viditelnosti, a tak když si schováme kotvy, oříznutí zůstává a v nosníku vzniká prázdný výřez viz obrázek 38.



Samozřejmě by bylo řešení, kdyby dutý tvar byl součástí vnořené kotvy a ta ořezávala nosník. To však také nefunguje, protože vnořená rodina do rodiny nemůže ořezávat rodinu do které je vložena.

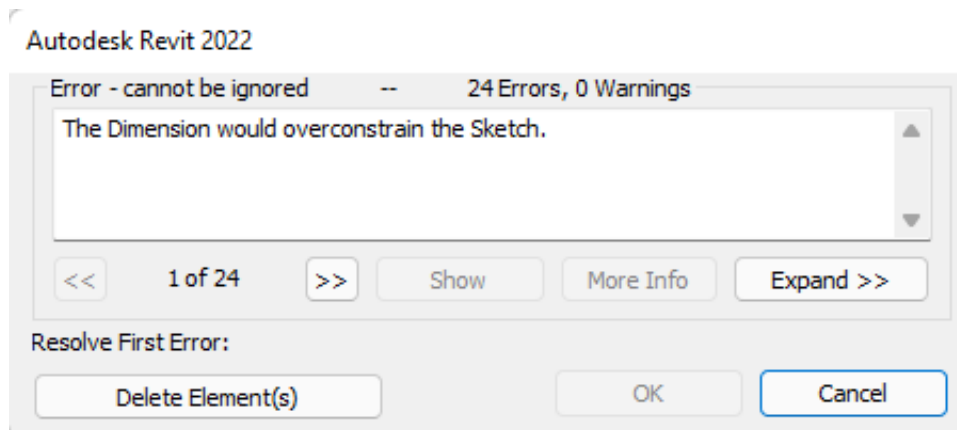
Další kotva, kterou používáme je spoj nosník sloup. Tato kotva je malinko složitější hlavně kvůli jejímu tvaru a obsahuje daleko více parametrů viz obrázek 25.



Obrázek 25 - Parametry kotvy nosník sloup

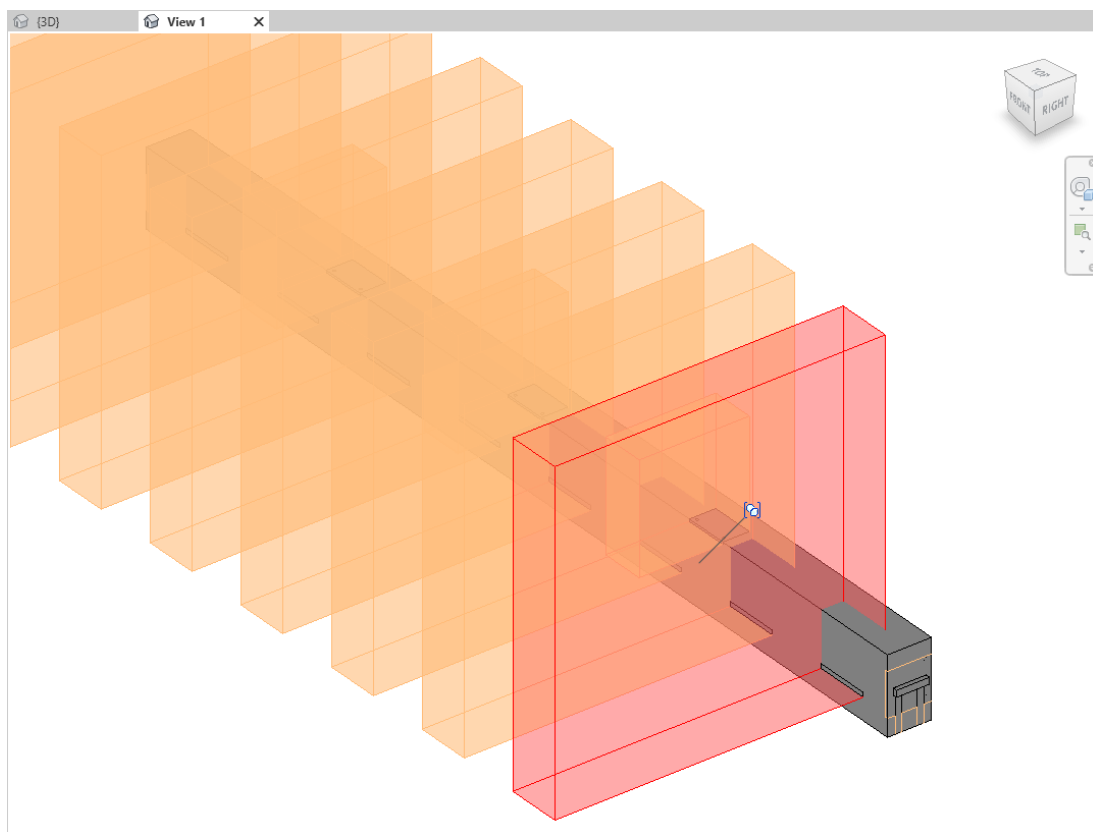
Dynamické chování kotev je simulované parametrem pole s matematickým vzorcem viz obrázek 18. Takto parametrizované množení kotev se jednotlivě vloží do skupiny a není možné upravovat pouze jeden prvek, aby nebyly upraveny i prvky ostatní na jednom nosníku. Zároveň není možné parametrizovat viditelnost skupiny prvků.

Problémy vzniklé při tvorbě nosníku byly mnohačetné. Největší překážkou byla parametrizace dutého tvaru, aby docházelo při změně rozměrů kotvy i ke změně rozměrů dutého tvaru. Dutý tvar jsem původně modeloval jako tažení, ale tato varianta nefungovala v případě, kdy byla přiřazena do skupiny, například v rámci kopírování kotev nosník spiroll. Chybu můžeme vidět na obrázku 26.



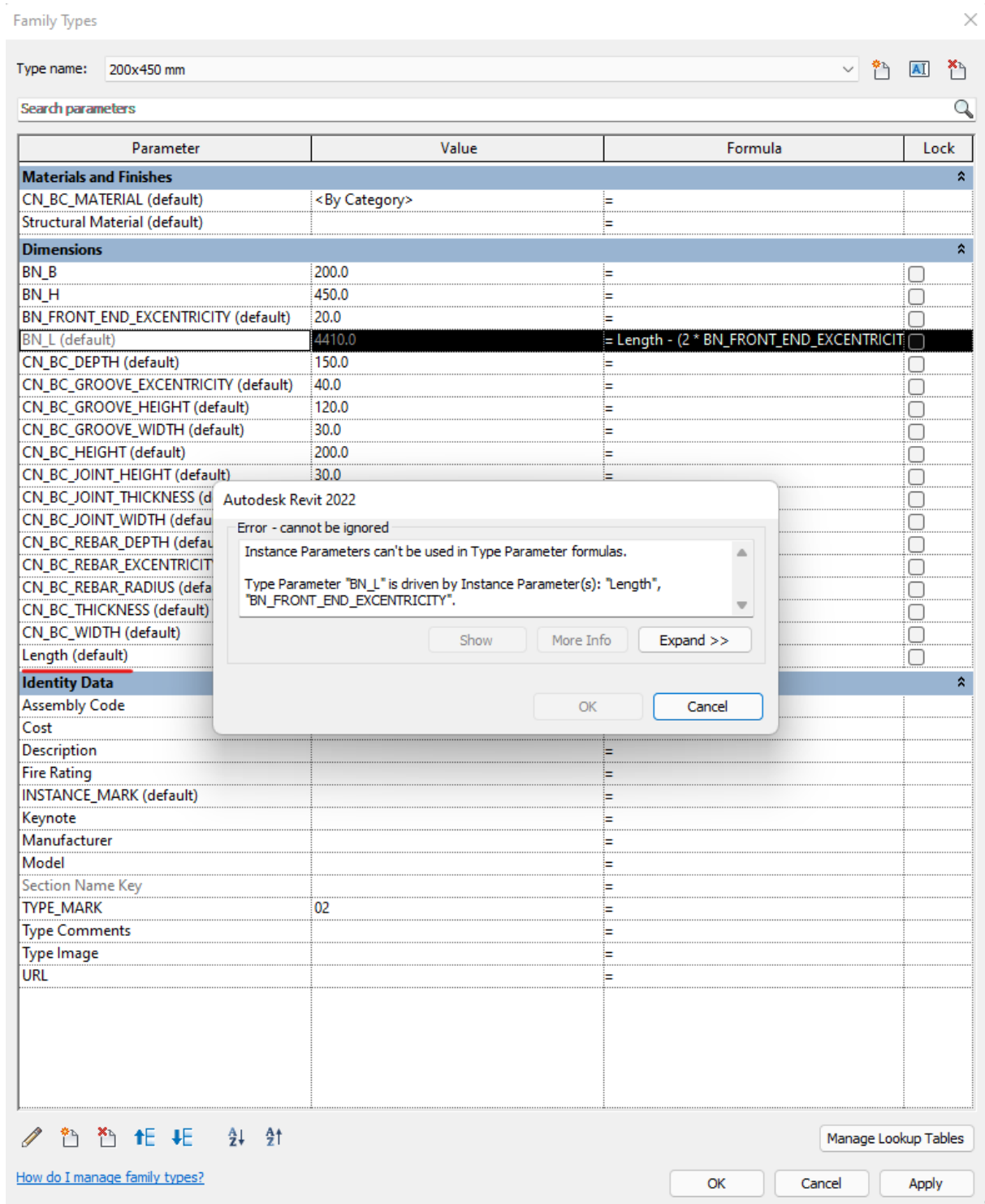
Obrázek 26 - Chyba při tvorbě dutého tvaru s vazbou na rozměry kotvy

Toto jsem vyřešil pouhou změnou modelovacího nástroje. Ve výsledku je tedy na nosníku použitý dutý tvar modelovaný jako tažení. Trajektorie tažení má referenční rovinu nastavenou jako spodní hranu kotev a konec i začátek úsečky je ukotven ke krajním stěnám kotvy, tato vazba se při úpravě rozměrů kotvy nepřeruší ale reaguje dynamicky. Následně je úsečce přiřazen nadrozměrný profil ořezu viz obrázek 27, který ořezává nosník. Profil je dutého tvaru má rozměr 1,5 x 1,5 m a nepředpokládáme, že by byl pro prvek systému nedostačující.



Obrázek 27 - Tvorba dutého tvaru pro ořez v místě kotev pro nosník Spiroll

Problém, který způsobovalo oříznutí prvku dutým tvarem, se projevoval pouze pokud jsem dutý tvar zařadil do skupiny. Pokud byl dutý tvar mimo skupinu a já jeho hranice náčrtu uzamkl k hranám kotvy, reagoval vždy dynamicky a Revit nehlásil chybu.



Obrázek 28 - Omezení použití parametrů v rovnici

### 3.4.3. Sloup systému DPS

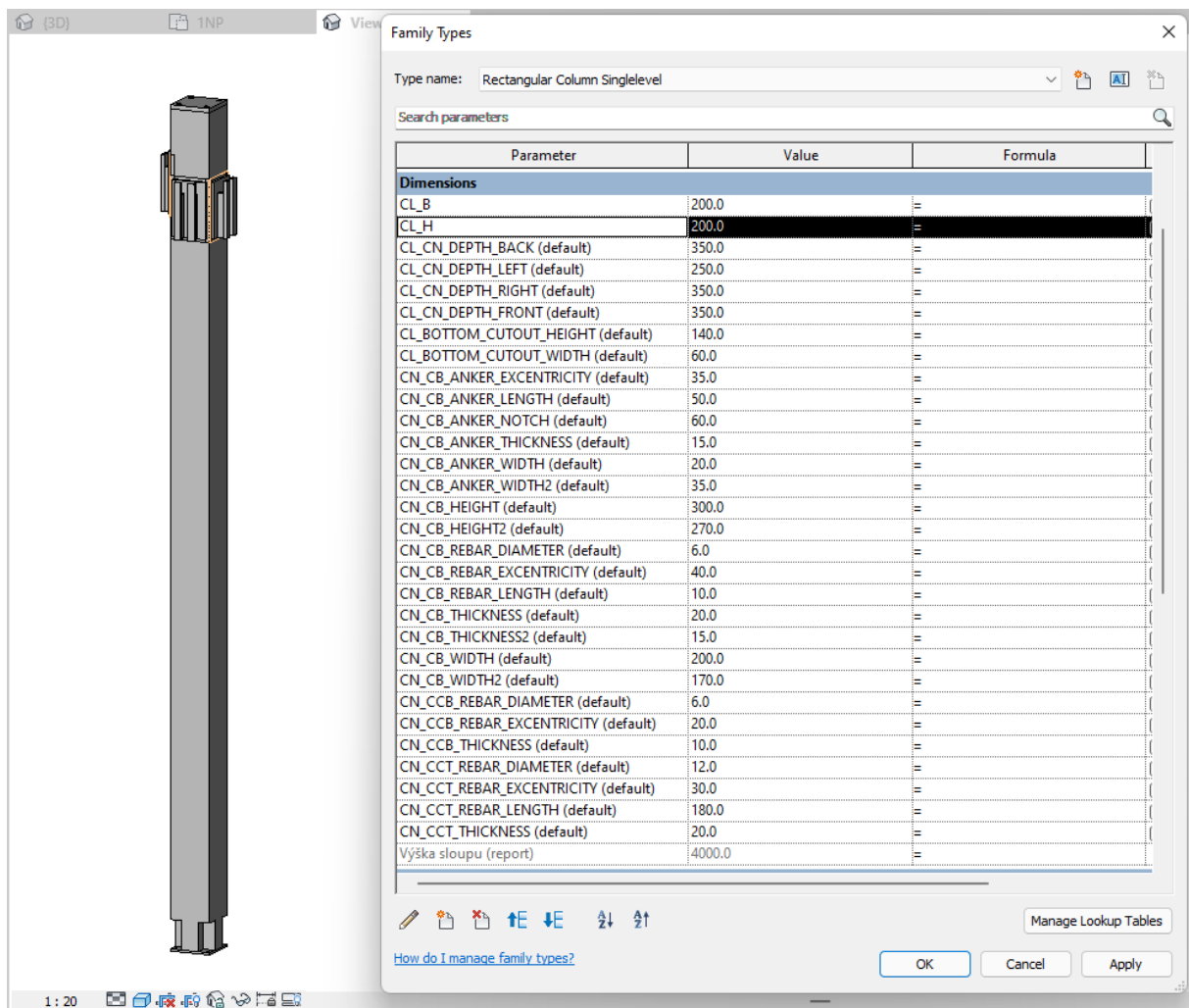
Rodiny sloupu jsou přizpůsobeny možnostem, kterými jsem omezen jak Revitem tak šablonou Structural Column, na které je rodina sloupu založena. Zase je nutné zmínit, že je možné sloup založit na šabloně rodiny obecný model a vyhnout se tak některým omezením, avšak to bychom následně přišli o analytický prvek modelu sloupu.

Sloup systému DPS je uvažován jako rodina přes jedno patro nebo přes patra dvě. Sloup se oproti nosníku liší taktéž v kopírování kotev. Jelikož máme sloup obdélníkový, hexagonální a oktogonální, tak je možné kotvy osadit na všechny možné strany. Kotva sloupu s nosníkem má tak nastavený parametr viditelnosti dle pozice a stejně tak má parametrizované jednotlivé výškové pozice. Tímto způsobem řeším i kotvy, které jsou v druhé výškové rovině, tedy u sloupu přes dvě patra.

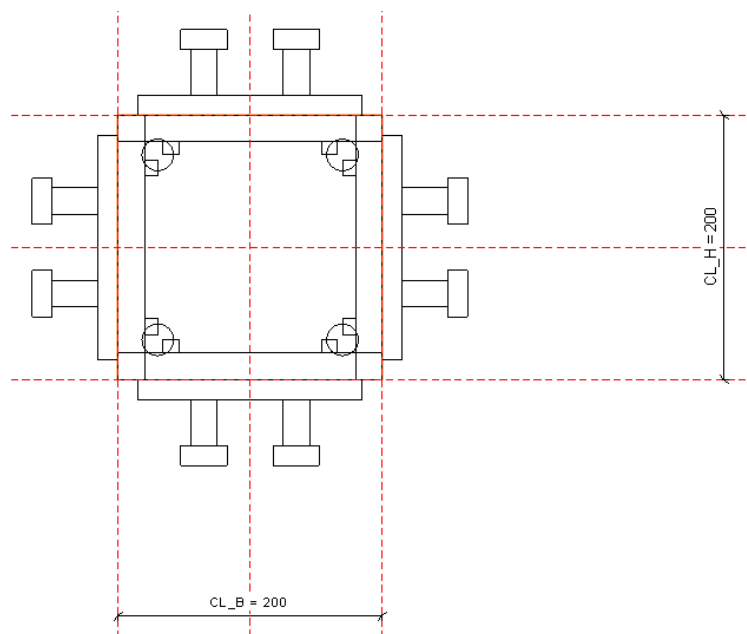
Pokud bychom chtěli mít jednu rodinu sloupu pro jedno a více pater, musel bych kopírování kotev zajistit stejným způsobem jako na nosníku, tedy funkcí pole, kterou bych parametrizoval. Problém však nastává, pokud by byl sloup přes jedno patro, s jednou řadou kotev, funkce pole totiž musí mít vždy 2 a více prvků, které kopíruje, jinak spouští chybu a uživateli nedovolí počet upravit. Kotvy jsou tedy ve sloupu parametrizovány přes parametr viditelnosti.

Rodina sloupu je tedy založená na šabloně konstrukční sloup. Běžně se v tomto případě parametrizuje pouze příčný řez sloupu, jelikož se výška sloupu modeluje od patra k patru nebo nastavením výškového odsazení horního nebo spodního. V našem případě jsou v rodině sloupy na víc pro spoj sloup nosník a pro spoj sloup sloup. Rodinu čtvercového sloupu přes jedno patro můžeme vidět na obrázku 29.

Parametrizace příčného profilu funguje stejně jako u nosníku, tedy za pomoci referenčních rovin, ke kterým se ukotví hrany hmoty sloupu viz obrázek 30. Pokud se jedná o sloup čtvercový, je toto jednoduché a jasné ale u sloupu s více hranami je postup trochu jiný, jak si ukážeme dále.



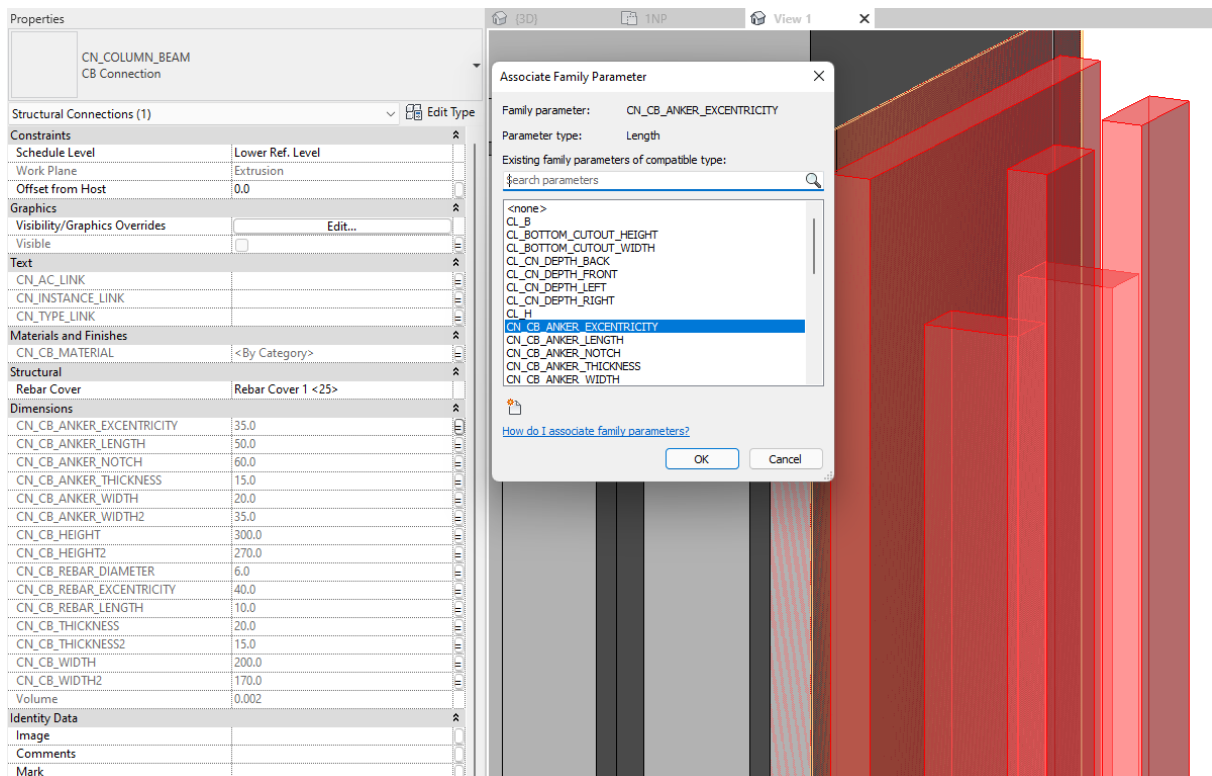
Obrázek 29 - Rodina sloupu přes jedno patro



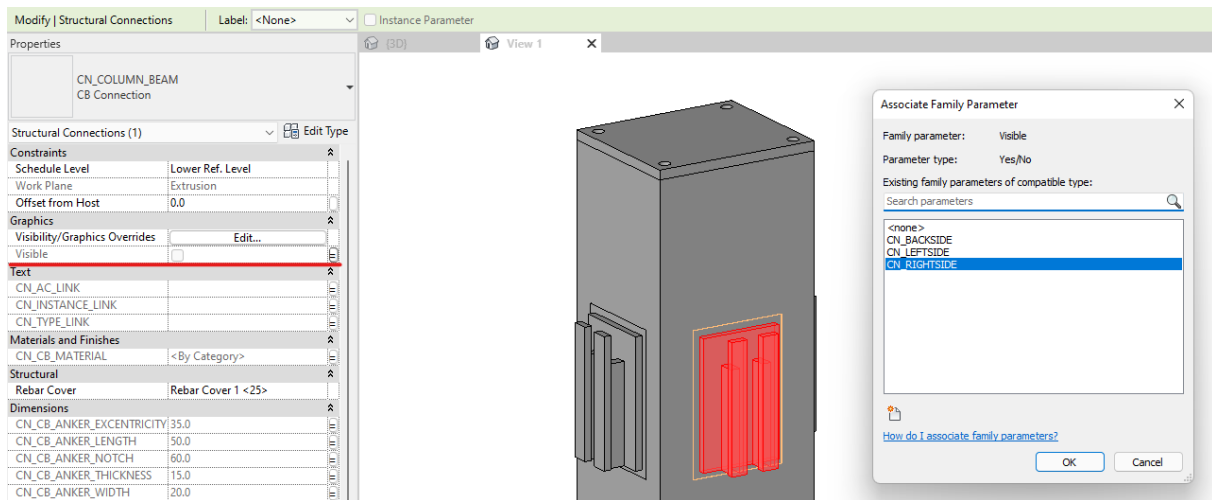
Obrázek 30 - Půdorysný pohled na rodinu sloupu

Co se týče kotev vložených do sloupu, stejně jako u nosníku, jsou tři. Spoj sloup sloup je rozdělený na horní a spodní, i když se jedná o identickou rodinu, tedy stejně parametrizovanou kotvu. Rozdíl pak může být v tom, že kotva na horním líci, může mít jiný, tvar a pozici kotevních prvků. Třetí kotvou je pak kotva sloup nosník. Taktéž by mohla být varianta s kotvami pro ztužující stěnu, kterou jsem v našem případě vynechal a použil ztužující stěnu pouze s kotvením do základového pasu a nosníku. Kotvy jsou stejně jako u nosníku vnořené rodiny, založené na kategorii obecný model a zaříděny do kategorie konstrukční spoj.

Kotvy mají taktéž parametry přiřazeny parametrům nově vytvořeným v rodině sloupu, se stejným jménem viz obrázek 31. Viditelnost kotev poté v tomto případě ovládáme parametrem viditelnosti, který je v šabloně rodiny jako vestavený parametr a je možné ho přiřadit každé plné hmotě nebo vnořené rodině viz obrázek 32. Dutému tvaru parametr viditelnosti přiřadit nelze.



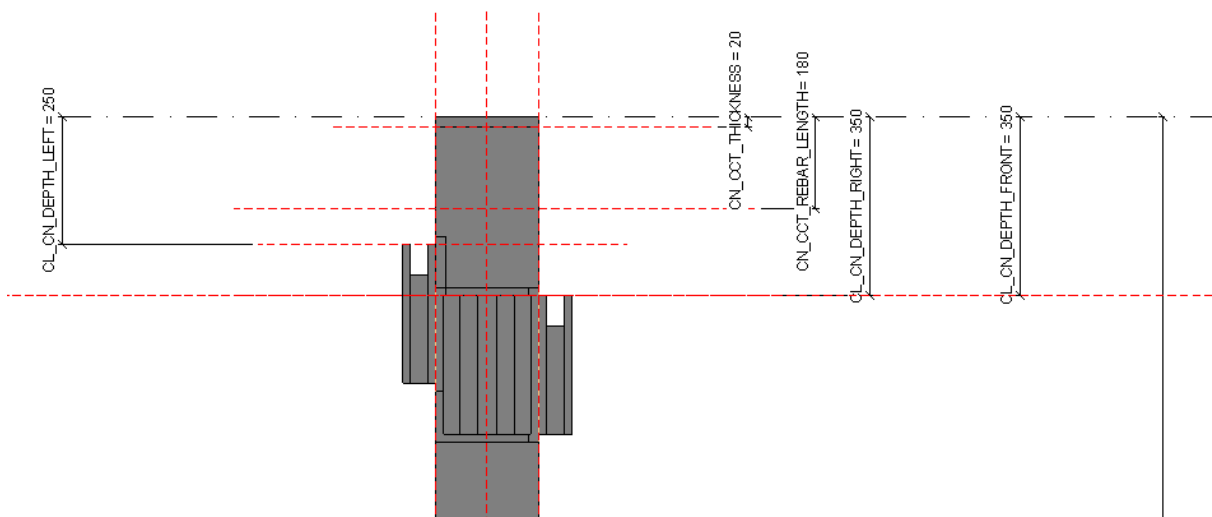
Obrázek 31 - Kotva sloup nosník a přiřazení parametru v nosníku



Obrázek 32 - Parametr viditelnosti pro kotvy sloup nosník

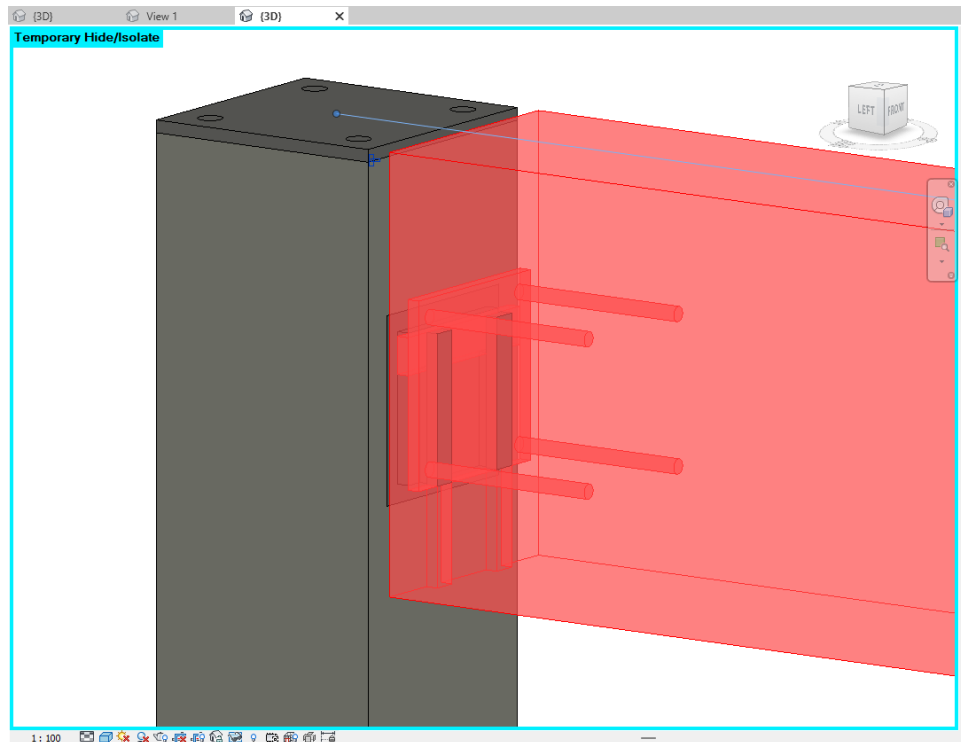
U sloupů, přes jedno patro, je parametr viditelnosti pro kotvy nastaven pro všechny kotvy, kromě kotvy v čele sloupu. Z toho důvodu, že sloup bude vždy mít alespoň jednu kotvu, a to právě tu čelní. Avšak u sloupu přes patra dvě, může být kombinace kotev daleko rozmanitější, a tak je parametr viditelnosti přiřazen všem kotvám v první i druhé řadě.

Samotné kotvy sloup nosník jsou poté ukotveny na hranu profilu sloupu a jeho střed. Jejich výšku ovládáme parametrem pro každou kotvu zvlášť. Pokud tedy máme sloup oktogonální přes dvě patra, máme zde 12 parametrů viditelnosti kotvy a 12 parametrů výškového odsazení kotev. Parametr výškového odsazení kotvy se taktéž liší, pokud se jedná o sloup přes jedno nebo dvě patra viz obrázek 33.



Obrázek 33 - Parametrizace pozice kotvy sloup nosník

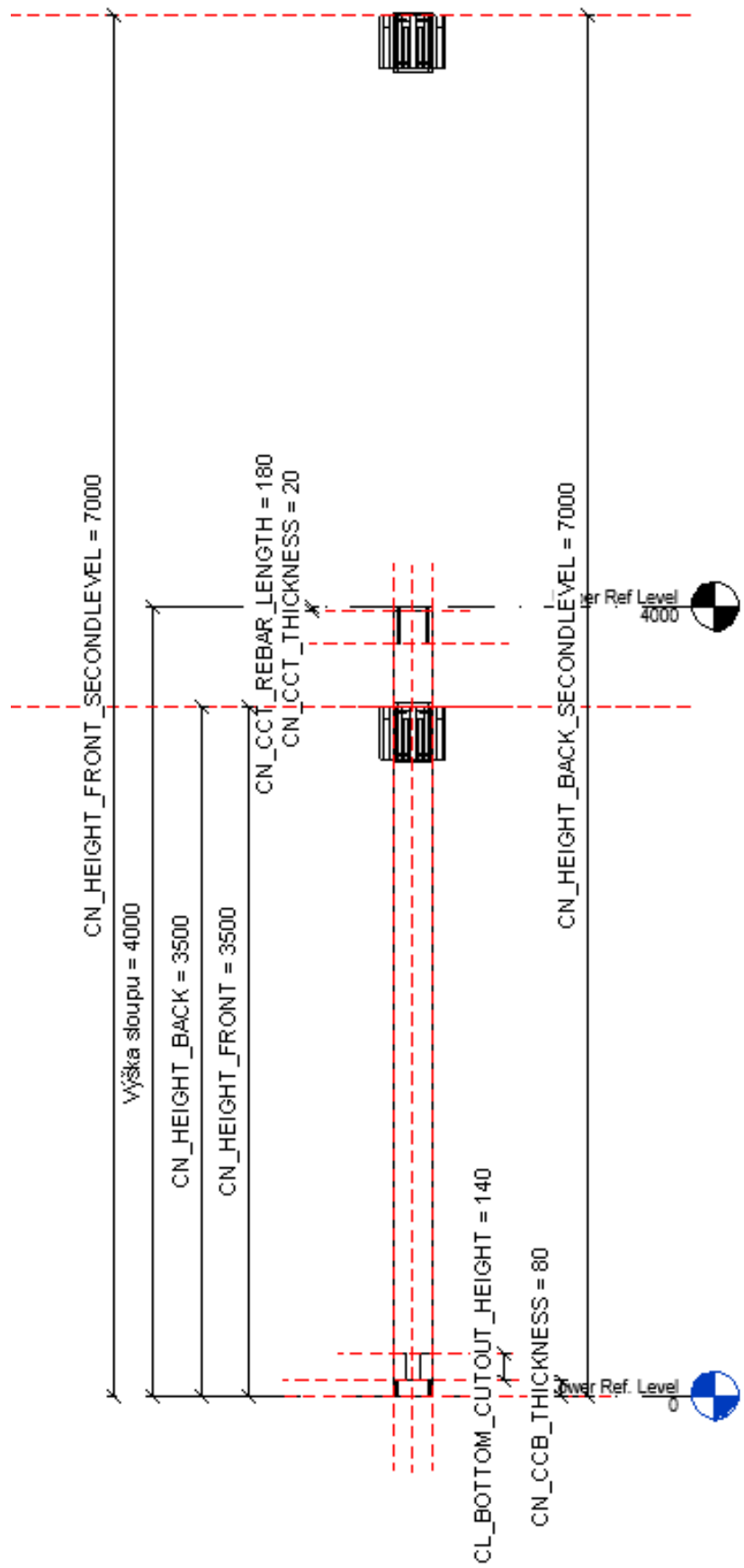
Kotvy sloup nosník, jsou parametrizovány na určitou výšku hrany stejně jako kotva nosník sloup použítá na nosníku. Pokud tedy parametr bude mít stejnou hodnotu jako parametr odsazení kotvy na nosníku, kotvy do sebe přesně zapadnou viz obrázek 34.



Obrázek 34 - Vazba kotvy sloup nosník a nosník sloup

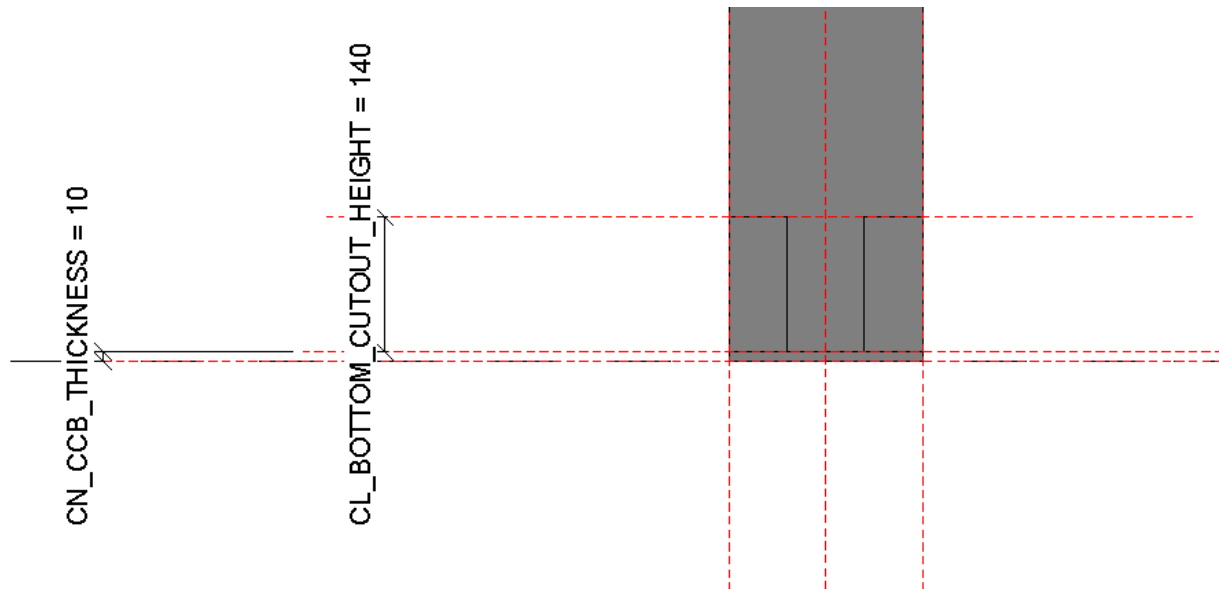
U sloupu přes dvě patra je poté výškové odsazení kotev parametrizováno od paty sloupu, viz obrázek 35, jelikož by nemělo přichycení parametru k horní hraně sloupu takový benefit jako u sloupu přes jedno patro. Tím pádem je uživatel nucen si výšku kotvy vzhledem ke kotvě nosník sloup spočítat.



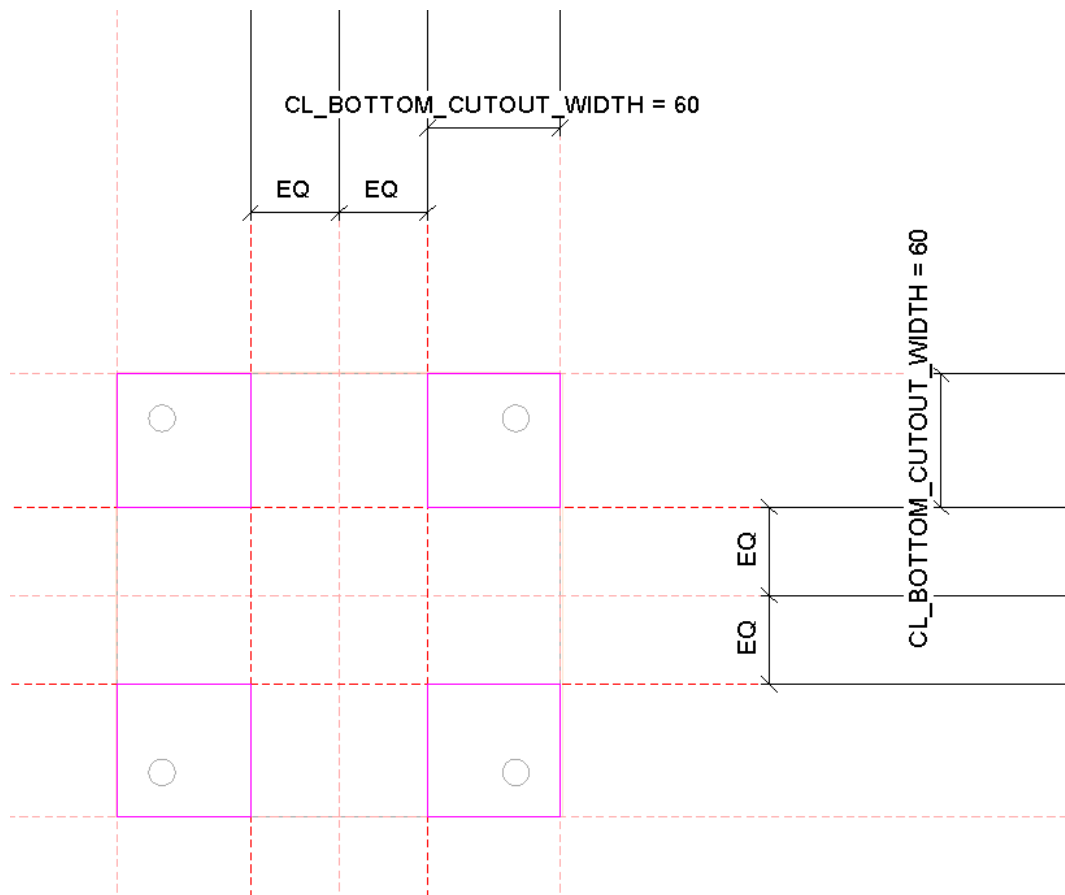


Obrázek 35 - Pohled na sloup přes dvě patra

Dále máme parametricky vytvořený dutý tvar, který ořezává prostor pro šrouby při montáži v patě sloupu viz obrázek 36 a 37.

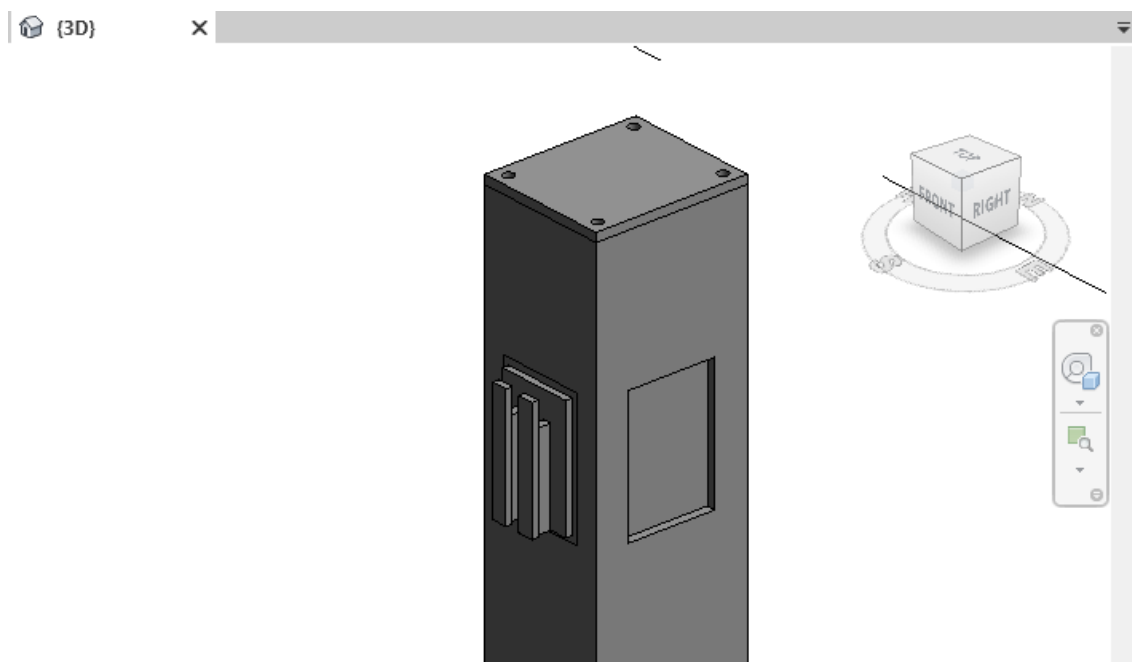


Obrázek 36 - Parametrizace dutého tvaru ořezání v patě sloupu



Obrázek 37 - Parametrizace dutého tvaru ořezání v patě sloupu půdorys

Ořezání sloupu v místě kotvy je v tomto případě modelováno jako dutý tvar tažením. Hraný tažení jsou ukotveny k hranám kotvy. Problém však vzniká, pokud v místě kotvy, kotva není, tedy nemá aktivní parametr viditelnosti ale dutý tvar toto místo i tak ořezává viz obrázek 38.



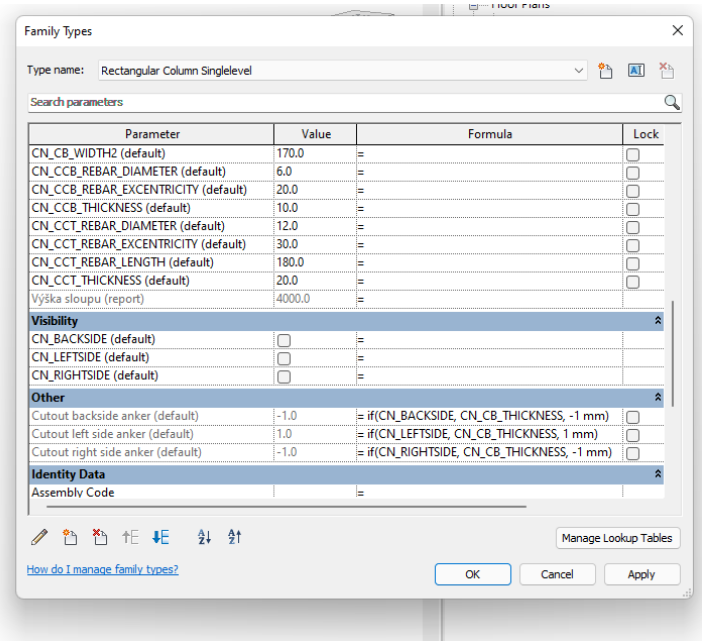
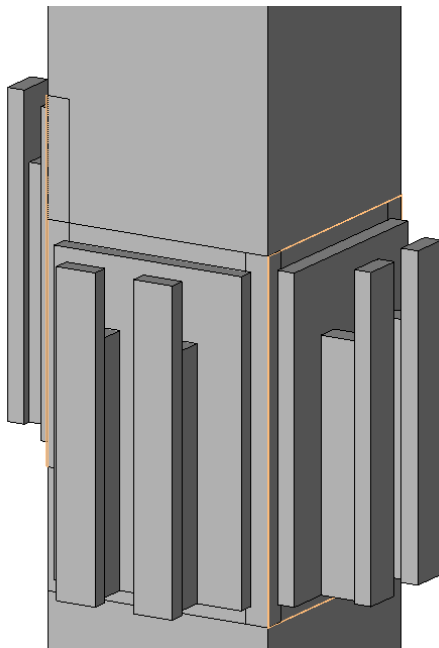
Obrázek 38 - Výřez v místě kotvy i přes její vypnutou viditelnost

Jak jsme psali výše, dutému tvaru nelze nastavit parametr viditelnosti ale lze mu nastavit v případě tažení parametr koncového a počátečního prodloužení. Tento parametr jsem složitě přes vzorce využil tím způsobem, že pokud je parametr viditelnosti aktivní, dutý tvar má koncové prodloužení rovné tloušťce kotvy. Pokud ale parametr viditelnosti aktivní není, dutý tvar má prodloužení rovné 1 mm vždy směrem z nosníku, a tedy sloup neořezává. Tyto parametry jsou v rodině pojmenovány jako Cutout backside anker viz obrázek 38.

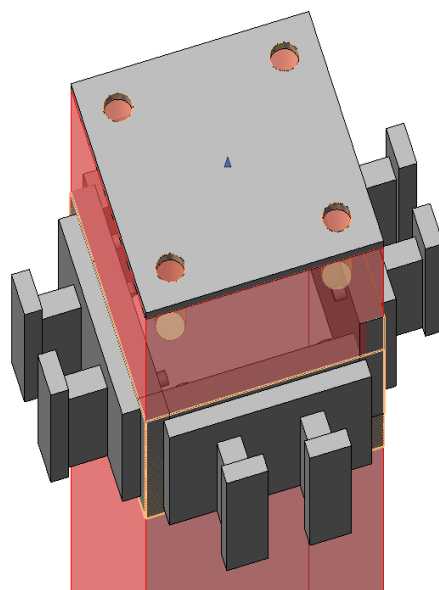
Pokud se jedná o kotvy spoj sloup sloup, které mají otvory pro závitové spojovací tyče, ty mají také přiřazeny dutý tvar, právě pro vyříznutí otvoru v místě spoje, závitovou tyčí. Dutý tvar mění poloměr a pozici v závislosti na parametrech kotvy. Hloubku zářezu pak ovládá uživatel parametrem vytvořeným v rodině sloupu viz obrázek 40. Zároveň je třeba upozornit na možnost ovládní viditelnosti kotvy sloup sloup horní a to z důvodu, že může být použit v posledním patře, bez potřeby spojení s další řadou sloupů. Pokud je viditelnost kotvy zapnuta, duté tvary ořezávají prostor pro závitové tyče. Pokud je viditelnost neaktivní, duté tvary mají prodloužení nastavené hodnotou 0 a nic neořezávají viz obrázek 39.

Visibility			
CN_CCT_VISIBILITY (default)	<input checked="" type="checkbox"/>	=	
CN_BACKSIDE (default)	<input type="checkbox"/>	=	
CN_LEFTSIDE (default)	<input type="checkbox"/>	=	
CN_RIGHTSIDE (default)	<input type="checkbox"/>	=	
Other			
CCT rebar length (default)	180.0	=	if(CN_CCT_VISIBILITY, CN_CCT_REBAR_LENGTH, 0 mm)
CCT visibility (default)	20.0	=	if(CN_CCT_VISIBILITY, CN_CCT_THICKNESS, 0 mm)
Cutout backside anker (default)	-1.0	=	if(CN_BACKSIDE, CN_CB_THICKNESS, -1 mm)
Cutout left side anker (default)	1.0	=	if(CN_LEFTSIDE, CN_CB_THICKNESS, 1 mm)
Cutout right side anker (default)	-1.0	=	if(CN_RIGHTSIDE, CN_CB_THICKNESS, -1 mm)
Identity Data			

Obrázek 39 - Viditelnost kotvy sloup sloup horní



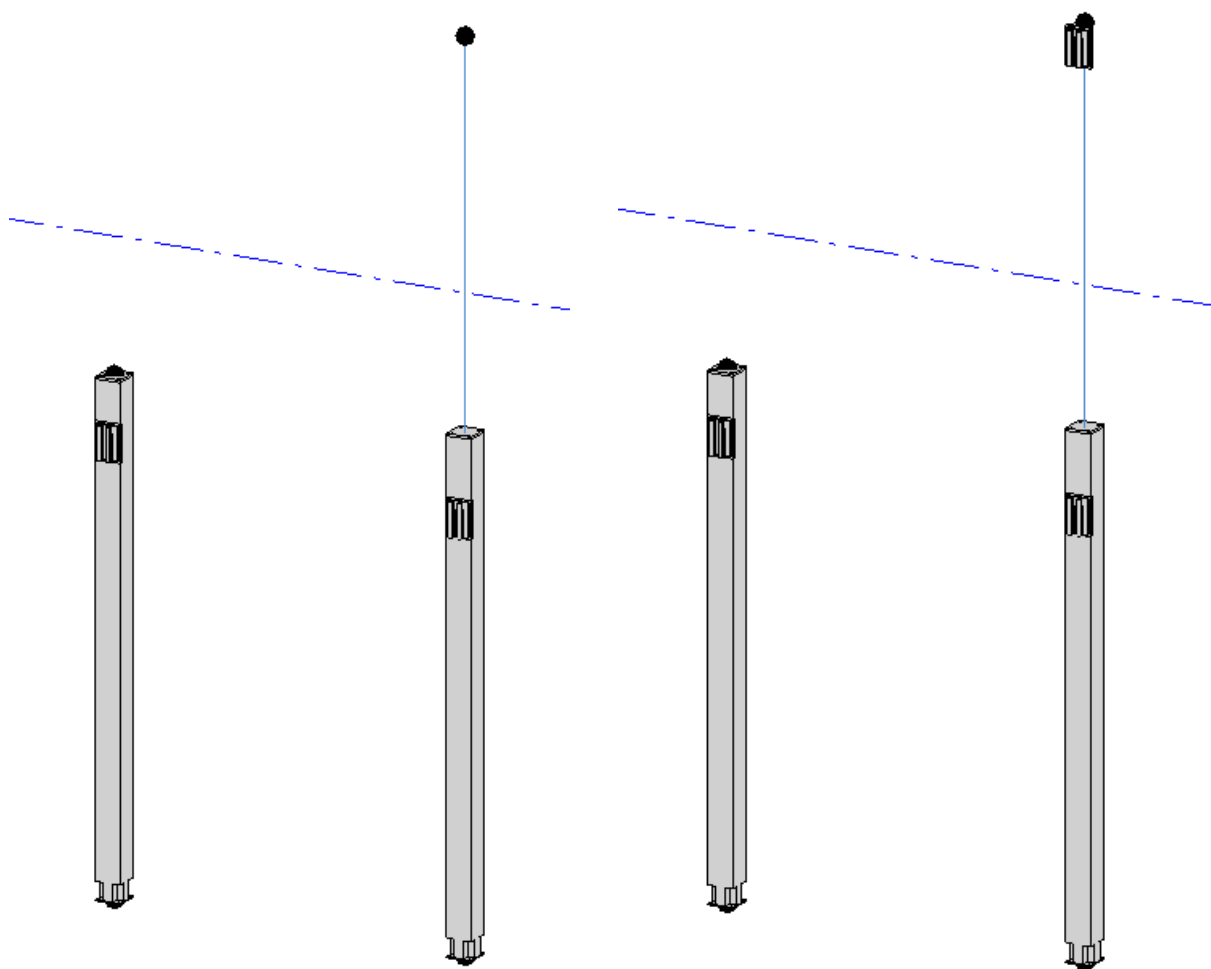
Obrázek 40 - Řešení problému výřezu



Obrázek 41 - 3D pohled na sloup a vazby k jeho ořezání

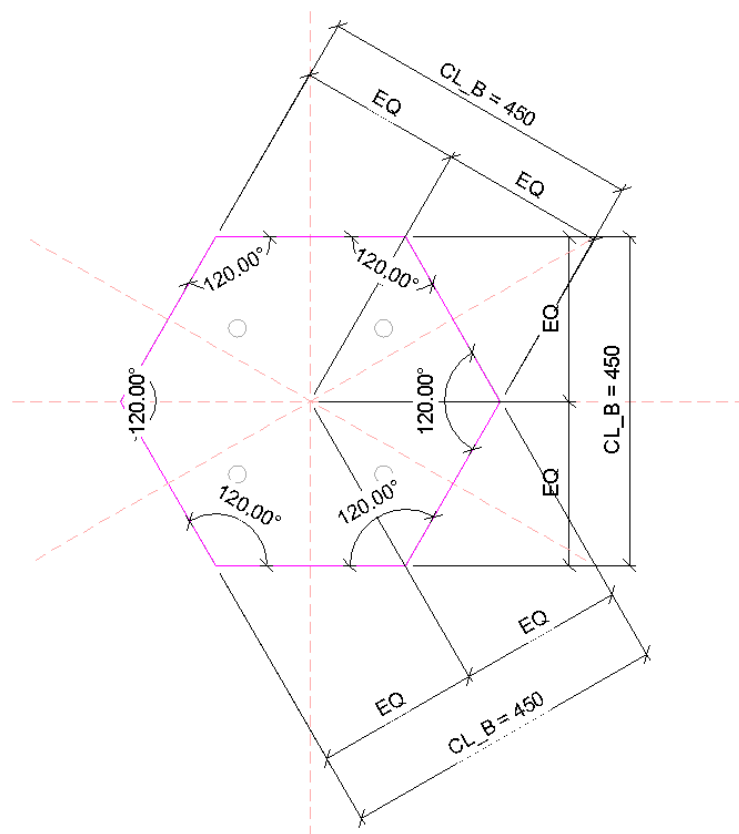
Co jsem zatím nevysvětlil je důvod dělení sloupu na rodinu přes jedno a dvě patra. Jelikož by teoreticky mohla sloužit pouze rodina sloupu přes dvě patra i pro sloup přes patro jedno vzhledem k přiřazeným parametrům viditelnosti jednotlivým kotvám. Avšak to nelze kvůli zachování analytického modelu. Při vložení rodiny do projektu byla grafická prezentace prvku perfektní. Problém nastal u modelu analytického. Jak si můžete na obrázku 42 všimnout, je nastavený parametr výšky kotev pro druhou řadu. Tato kóta nebo parametr u sloupu v modelu indikuje, že se v místě nachází analytický bod a vzniká tak rozpor mezi grafickým a analytickým modelem.

Výše zmíněný problém byl právě vyřešen rozdělením rodiny sloupu na dvě, tedy rodinu sloupu pro 1 patro a rodinu sloupu pro 2 patra. Důvod tohoto rozdělení je i fakt, že pomocí příkazu pole, kterým modeluji kopírování kotev v nosníku, je podmíněno minimálně 2 a více prvky v poli. Není tedy možné mít tímto způsobem parametrizovaný i sloup o jedné výšce kotev.



Obrázek 42 - Rozdíl mezi sloupem přes jedno a přes dvě patra

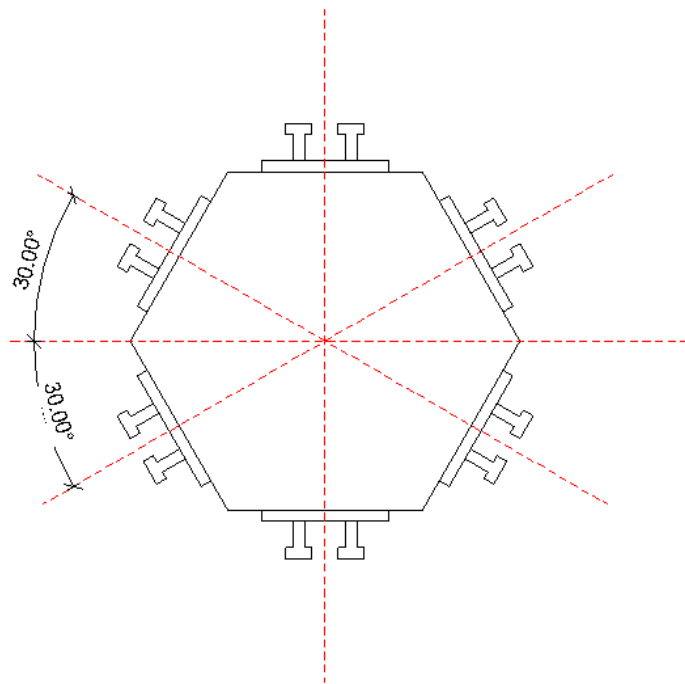
Sloup ve tvaru hexagonu tedy šestihran, je stejná rodina jako rodina sloupu obdélníkového. Jediný rozdíl je pak v počtu kotev a parametrizace příčného profilu. Výjimečně se v tomto případě nevyužívají referenční rodiny ale profil se parametrizuje přímo v editoru. Kotvy se poté přichycují k náčrtu profilu. V tomto případě taktéž musíme parametrizovat nejen šířku profilu ale je třeba uzamknout přesnou hodnotou úhel mezi hranami viz obrázek 43.



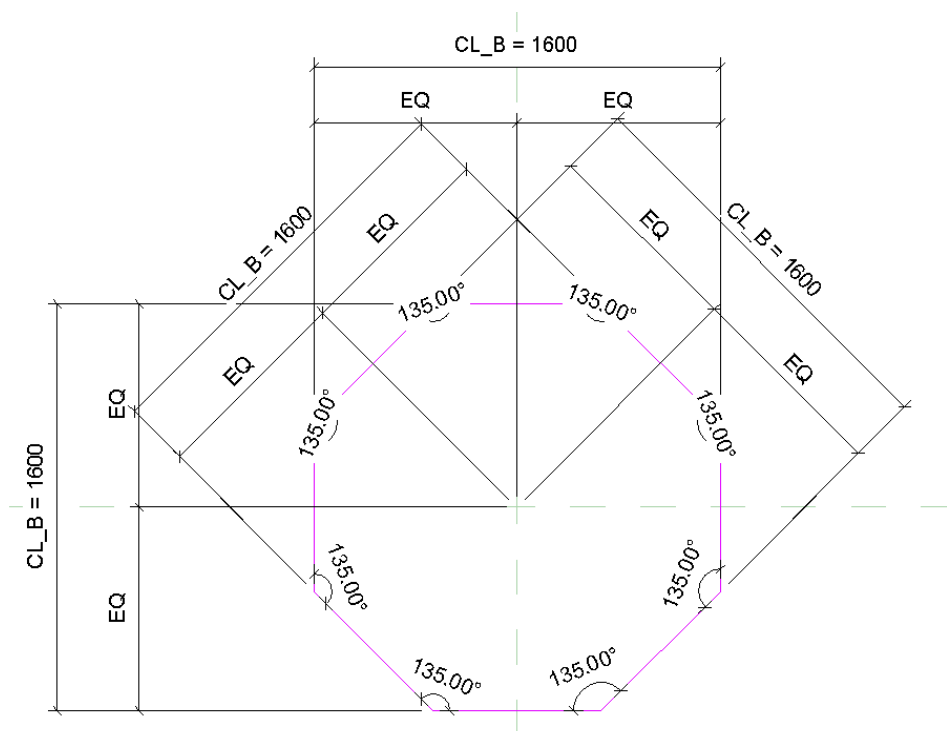
Obrázek 43 - Parametrizace sloupu ve tvaru šestihranu

Pro uchycení kotev vždy na střed hrany sloupu jsem si poté pomohl pevným ukotvením referenčních rovin a následnému přichycení středu kotvy k této rovině viz obrázek 44.

Sloup ve tvaru oktagonu tedy osmihran se modeluje a parametrizuje stejně jako sloup hexagonální s tím rozdílem, že musím ukotvit a parametrizovat další dvě rovnoběžné hrany oproti hexagonu viz obrázek 45.

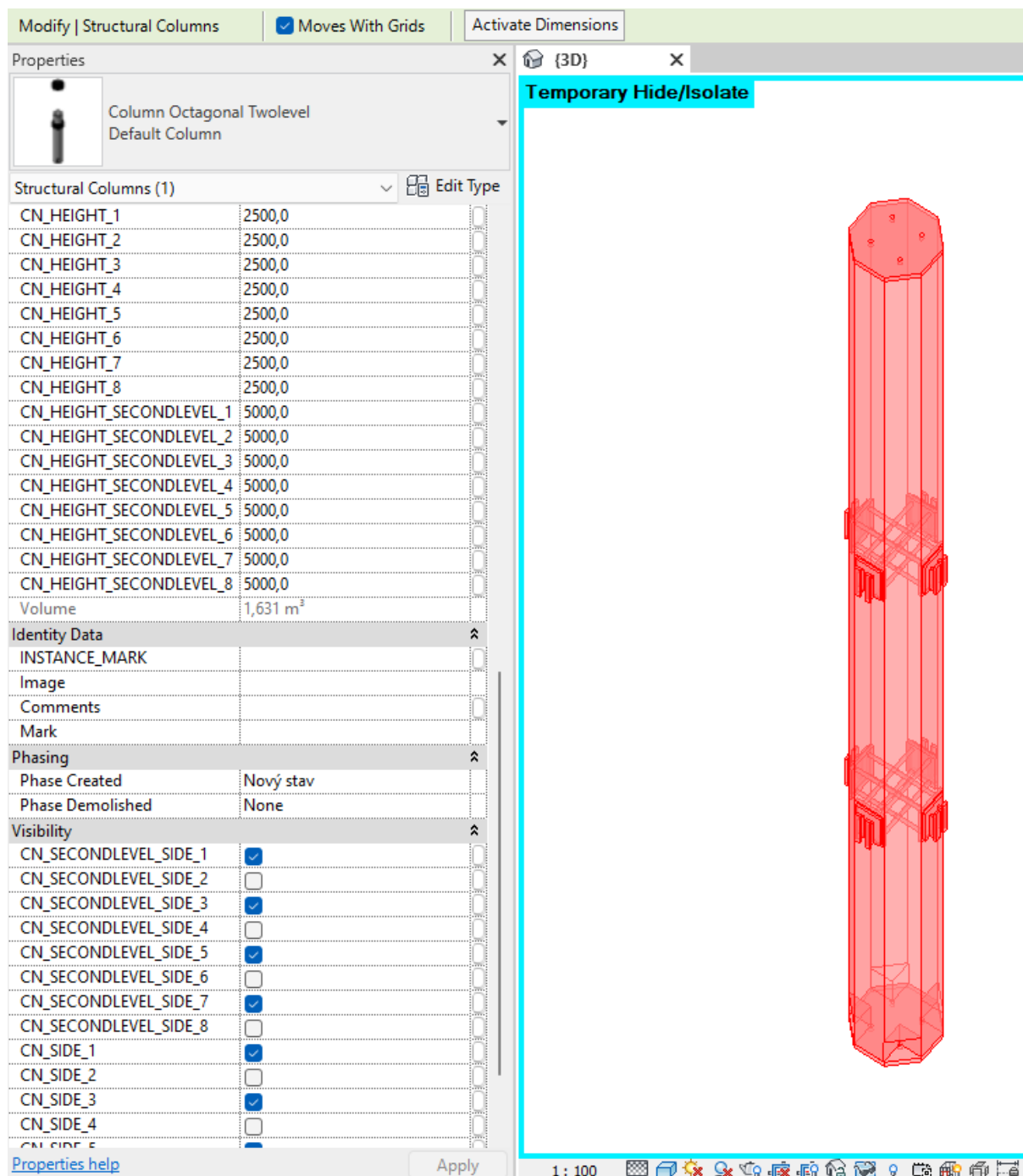


Obrázek 44 - Vazba kotev na sloupu ve tvaru šestihranu



Obrázek 45 - Parametrizace sloupu ve tvaru osmihranu

Stěny sloupu jsou označeny čísly od 1-6 nebo od 1-8. V katalogu prvků je pozice číslování u sloupu znázorněna. Výsledná rodina osmihranového sloupu s parametry viditelnosti a jejich výškové odsazení můžeme vidět na obrázku 46.

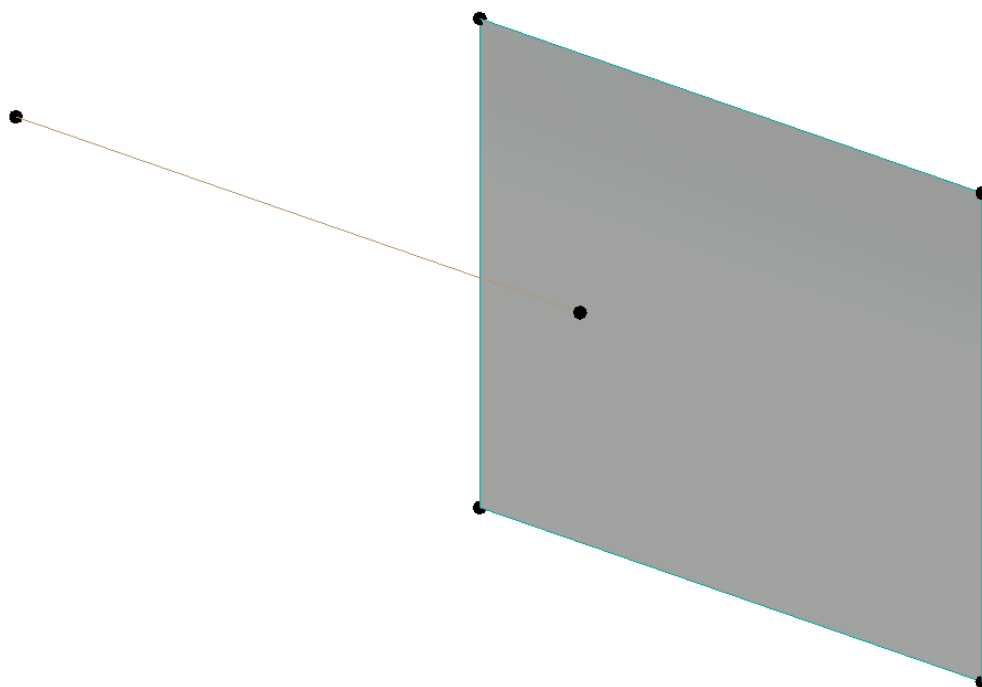


Obrázek 46 - Představení rodiny sloupu ve tvaru šestihranu



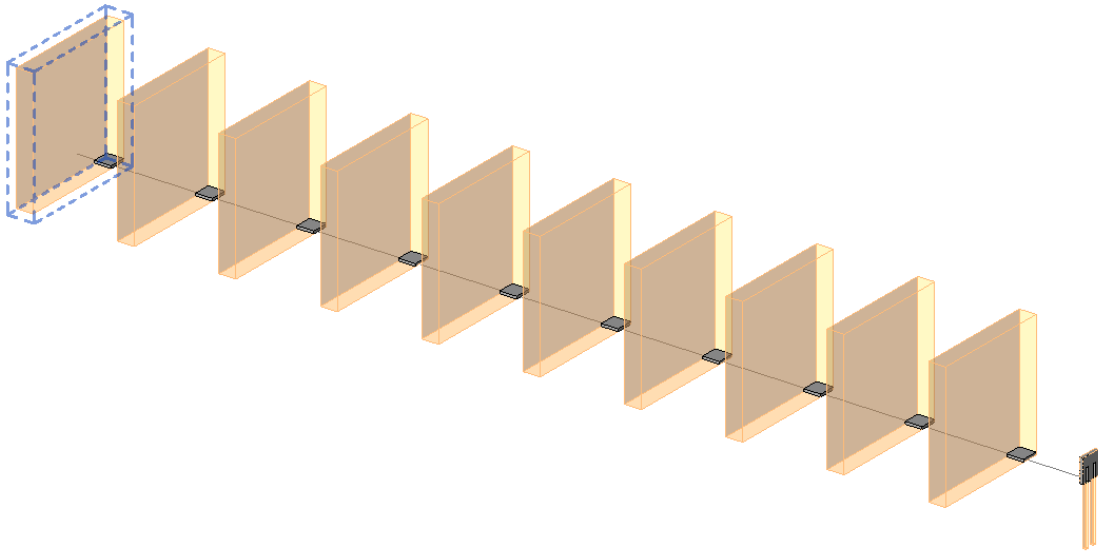
### 3.4.4. Stěna nosná

Stěny v Revitu se řadí mezi rodiny systémové. Není tak možné založit rodinu na šabloně konstrukční stěna jako nosník nebo sloup. Je možné založit rodinu na šabloně jiné ale následně není možné jí přiřadit kategorii stěny, jako je tomu například u kotev. Vždy je však možné založit rodinu stěny na šabloně například obecný model a kategorii jí nepřirázovat. V tomto případě jsme schopni stěnu parametrizovat podobně jako nosník. V tomto případě nám ale vzniká problém u analytického modelu. Jelikož obecný model neobsahuje analytický model, jednalo by se pouze o rodinu s grafickým zobrazením, která by si nesla určité informace. V analytickém modelu by se však neprojevila. Taktéž je možné založit rodinu stěny na šabloně konstrukční rámové konstrukce, nic méně tento způsob je taktéž z hlediska analytického modelu nepřijatelný. I když by graficky stěna vypadala jako stěny, analytický model by odpovídal ose se dvěma styčníky. Nicméně analytický model stěny, vypadá jinak viz obrázek 47, kde je rozdíl naznačen.



Obrázek 47 - Rozdíl mezi analytickým modelem nosníku a stěny

Řešením by byla tvorba rodiny duplicitní s rodinou nosníku, akorát bez hmoty, pouze s kotvami a dutými tvary viz. obrátek 48. Stěna by se poté modelovala na dvakrát, tedy pomocí systémové rodiny stěny s příslušnou tloušťkou, výškou a délkou a následně přes horní hranu vložena rodina s kotvami a dutými tvary. Příkazem spoj geometrii by došlo k propojení rodin a ořezání stěny dutým tvarem.



Obrázek 48 - Rodina výřezu pro stěnu

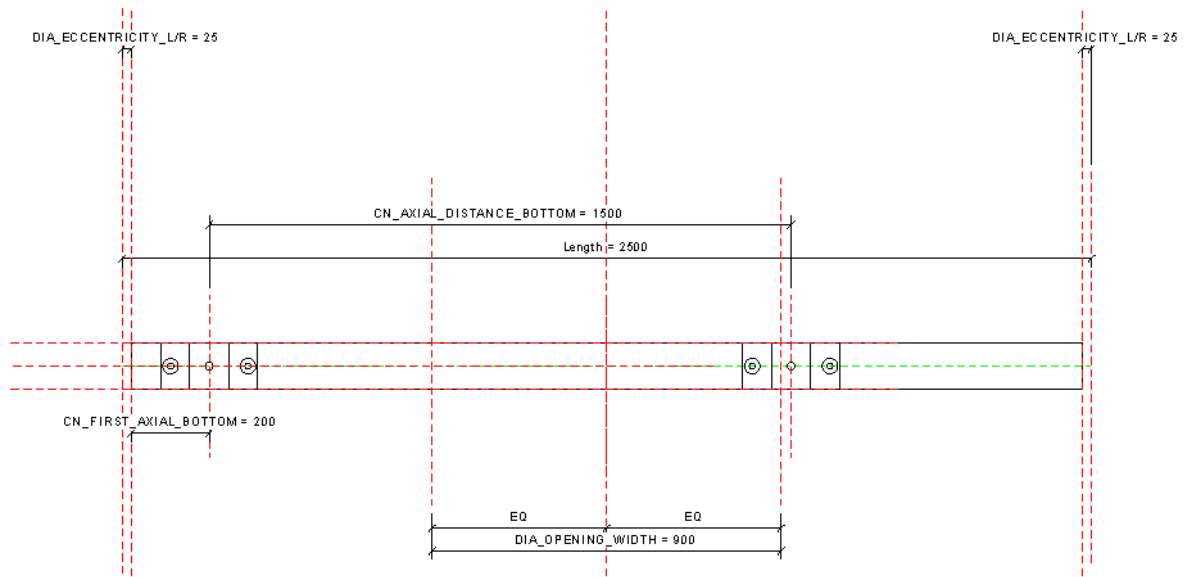
Druhým řešením je, se v tomto případě vzdát analytického modelu, vytvořit rodinu založenou například na šabloně metrický obecný model liniový a zatřídit stěnu do kategorie konstrukční rámová konstrukce. Tímto způsobem bychom alespoň dodrželi model grafický, ale následně by byla nutná úprava modelu analytického v softwaru pro konstrukční analýzu.

Jelikož to není nemožné ale vzhledem k našemu cíli, a to převést model do softwaru na statickou analýzu komplikující, rozhodli jsme se stěnu v tomto případě nezapojit do systému a referenční objekt bude vytvořen pouze z nosníků, sloupů a základových konstrukcí.

#### 3.4.5. Stěna ztužující systému DPS

Ztužující stěna, diafragma čelí stejnému problému jako stěna nosná. V tomto případě ale bude stěna modelována jako konstrukční ztužení a analytický model bude ve statickém softwaru nahrazen dvojicí táhel. Tedy rodina bude založena na šabloně obecný model liniový a bude mu přiřazena kategorie konstrukční ztužení.

Půdorysný pohled rodiny ztužující stěny můžeme vidět na obrázku 49. Jeho parametry poté na obrázku 50.



Obrázek 49 - Půdorysný pohled na rodinu ztužující stěny DPS

Family Types

Type name: Default Diaphragm

Search parameters

Parameter	Value	Formula	Lock
<b>Constraints</b>			
Default Elevation	0.0	=	<input type="checkbox"/>
Length (default)	2500.0	=	<input type="checkbox"/>
<b>Materials and Finishes</b>			
CN_DIA_MATERIAL (default)	<By Category>	=	<input type="checkbox"/>
Structural material (default)	<By Category>	=	<input type="checkbox"/>
<b>Dimensions</b>			
DIA_THICKNESS	120.0	=	<input type="checkbox"/>
DIA_HEIGHT	2900.0	=	<input type="checkbox"/>
DIA_EXCENTRICITY_L_R	25.0	=	<input type="checkbox"/>
DIA_EXCENTRICITY_T_B	50.0	=	<input type="checkbox"/>
DIA_LENGTH (default)	2450.0	= Length - 2 * DIA_EXCENTRICITY_L_R	<input type="checkbox"/>
DIA_DOOR_OPENING	<input type="checkbox"/>	=	<input type="checkbox"/>
DIA_OPENING_HEIGHT	2100.0	=	<input type="checkbox"/>
DIA_OPENING_WIDTH	900.0	=	<input type="checkbox"/>
CN_FIRST_AXIAL_BOTTOM (default)	200.0	=	<input type="checkbox"/>
CN_AXIAL_DISTANCE_BOTTOM (default)	1500.0	=	<input type="checkbox"/>
CN_FIRST_AXIAL_TOP (default)	200.0	=	<input type="checkbox"/>
CN_AXIAL_DISTANCE_TOP (default)	1500.0	=	<input type="checkbox"/>
CN_DIA_CUTOUT_WIDTH (default)	100.0	=	<input type="checkbox"/>
CN_DIA_CUTOUT_HEIGHT (default)	150.0	=	<input type="checkbox"/>
CN_DIA_ANKER_DIAMETER (default)	10.0	=	<input type="checkbox"/>
CN_DIA_LENGTH (default)	250.0	=	<input type="checkbox"/>
CN_DIA_REBAR_DIAMETER (default)	8.0	=	<input type="checkbox"/>
CN_DIA_REBAR_DIAMETER2 (default)	20.0	=	<input type="checkbox"/>
CN_DIA_REBAR_EXCENTRICITY (default)	100.0	=	<input type="checkbox"/>
CN_DIA_REBAR_LENGTH (default)	250.0	=	<input type="checkbox"/>
CN_DIA_REBAR_THICKNESS (default)	6.0	=	<input type="checkbox"/>
CN_DIA_THICKNESS (default)	25.0	=	<input type="checkbox"/>
<b>Other</b>			
CN_DIA_COUNT_BOTTOM (default)	2	= Length / CN_AXIAL_DISTANCE_BOTTOM	<input type="checkbox"/>
CN_DIA_COUNT_TOP (default)	2	= Length / CN_AXIAL_DISTANCE_TOP	<input type="checkbox"/>
Door opening (default)	-61.0	= if(DIA_DOOR_OPENING, DIA_THICKNESS / 2, (-DIA_THICKNESS / 2 - 1 mm))	<input type="checkbox"/>
<b>Identity Data</b>			

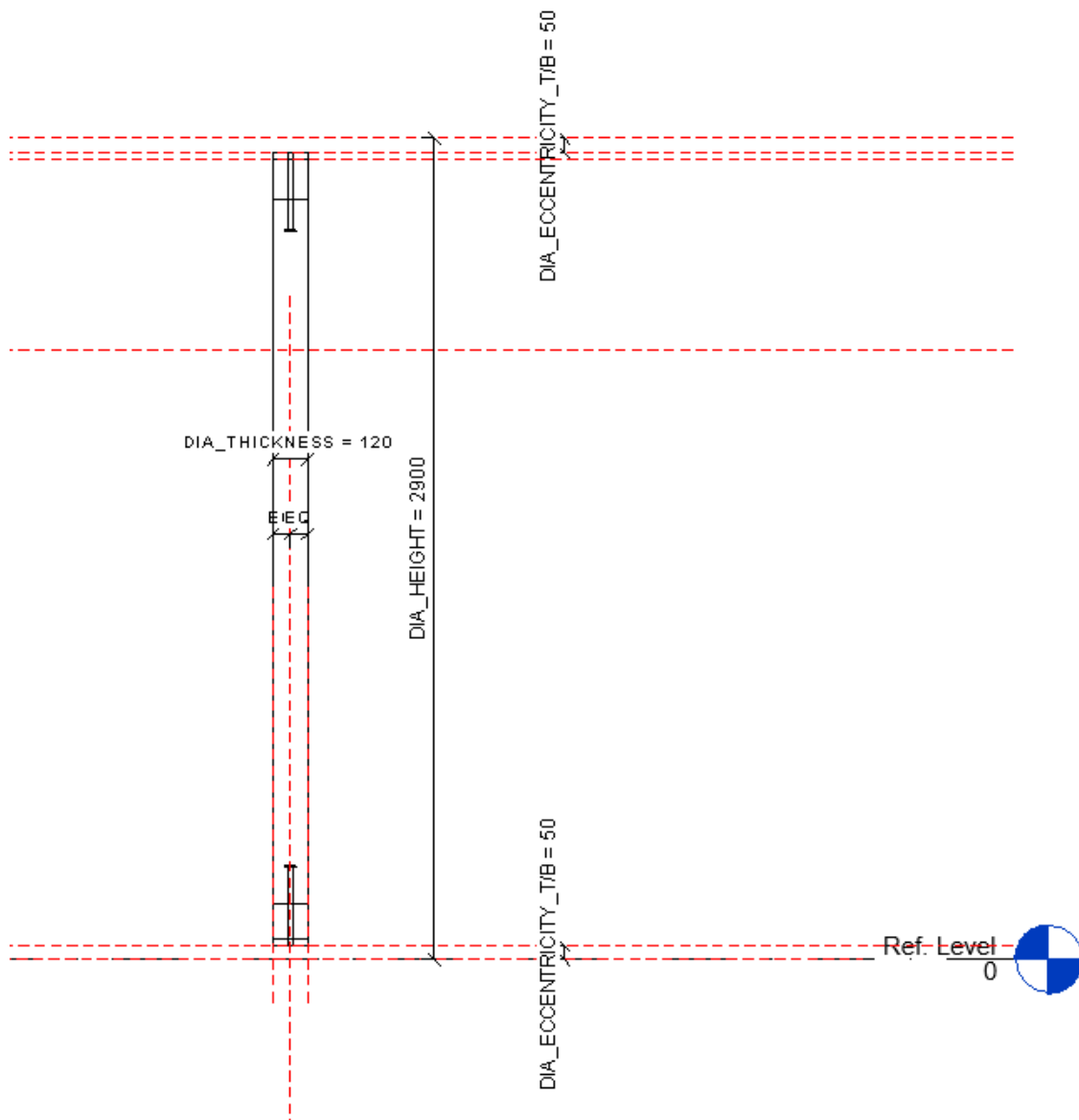
Manage Lookup Tables

OK Cancel Apply

Obrázek 50 - Parametry ztužující stěny DPS

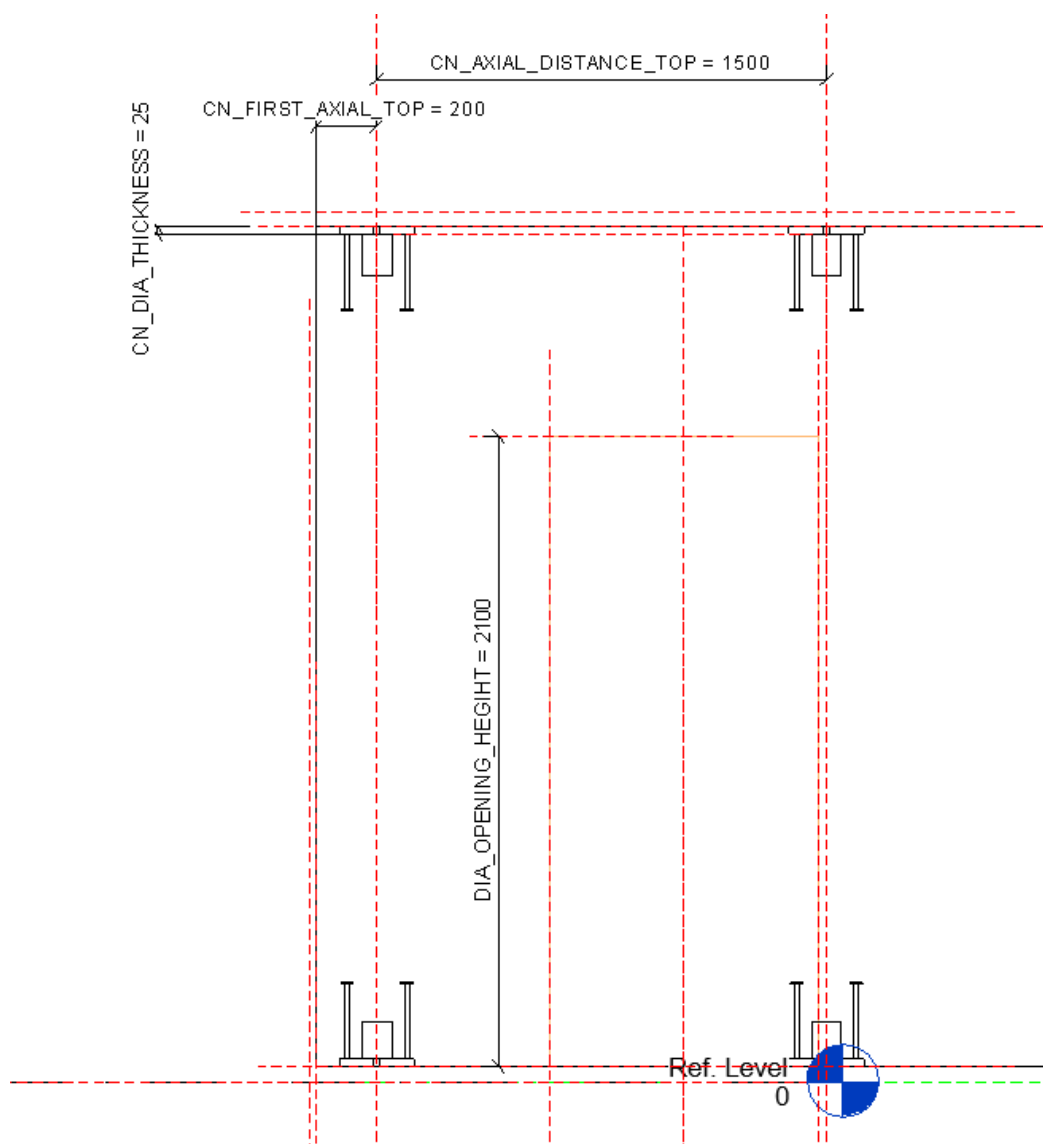
U ztužující stěny jsou použity kotvy na jejím horním a spodním okraji. V systému DPS jsou ale i varianty s kotvami na stranách stěny anebo na všech čtyřech stranách. Pro demonstraci jsme nemodelovali všechny varianty a vznikly tak i rodiny nosníků s kotvami pro ztužující stěnu, jak jsem psal výše.

Na obrázku 51 můžete vidět rovinu označenou jako Ref. level, neboli referenční rovinu. Tato vestavěná rovina určuje, k jaké referenční rovině se bude stěna modelovat. Jelikož se jedná o rodinu vytvořenou na šabloně obecný model liniový, je možné jí v modelu modelovat buď na referenční rovinu jako například podlaží nebo modelovat na pohled. Stěny se tedy modelují liniově v jejich patě a nastavujeme jí výšku a následně koncové, počáteční a horní, dolní odsazení.

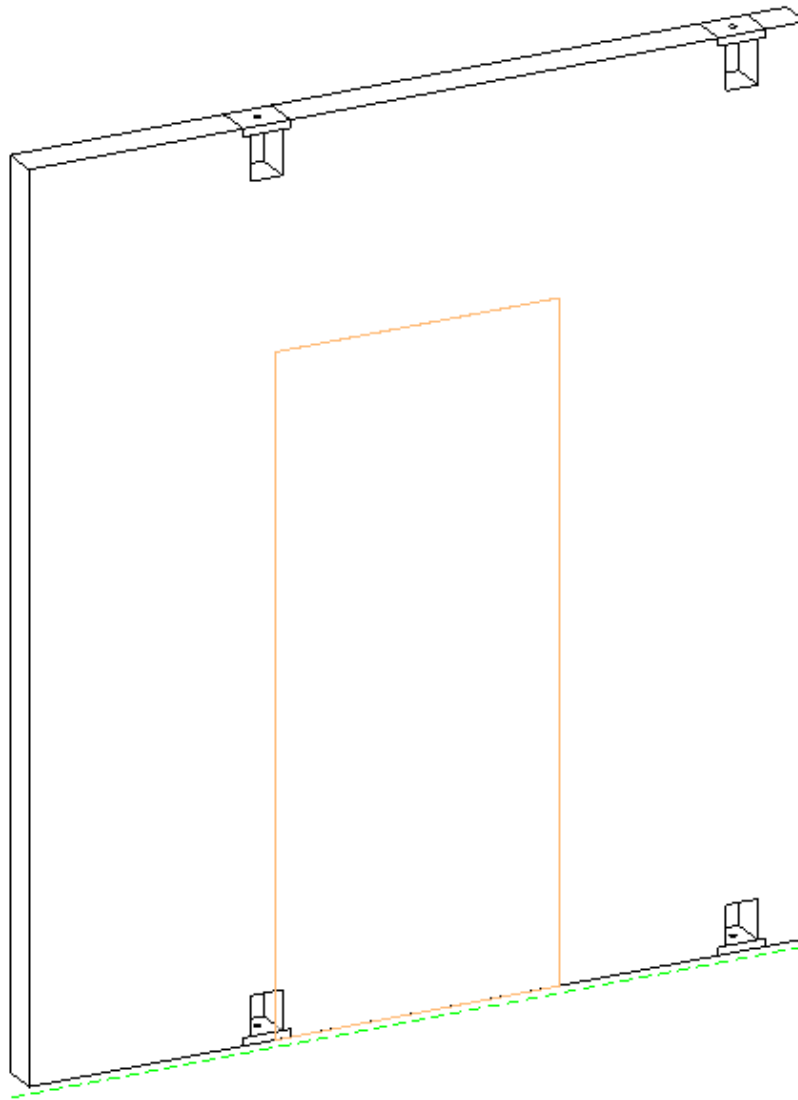


Obrázek 51 - Boční pohled na rodinu ztužující stěnu DPS

Problém, kterým jsem se musel zabývat byl výřez pro dveřní otvor. Jelikož, jak už víme nelze dutému tvaru přiřadit parametr viditelnosti, je toto řešeno podobně jako u výřezu pro kotvy na sloupu. Je nastavený parametr viditelnosti DIA\_DOOR\_OPENING, ale není přiřazen žádnému prvku nebo hmotě v rodině. Avšak pokud je aktivní, vzorec v parametru Door opening způsobí, že je koncové prodloužení rovno polovině tloušťky stěny. Počáteční prodloužení je ukotveno k opačné hraně stěny a druhá hrana je tak ovládána tímto vzorcem. Pokud je však parametr viditelnosti neaktivní, vzorec nastaví dutému tvaru konečné prodloužení hodnotu, která vymodeluje tvar opačným směrem od stěny a přičte 1 mm. Tedy dutý tvar vystupuje ven ze stěny a nic neořezává viz obrázek 53.



Obrázek 52 - Čelní pohled na rodinu ztužující stěny DPS

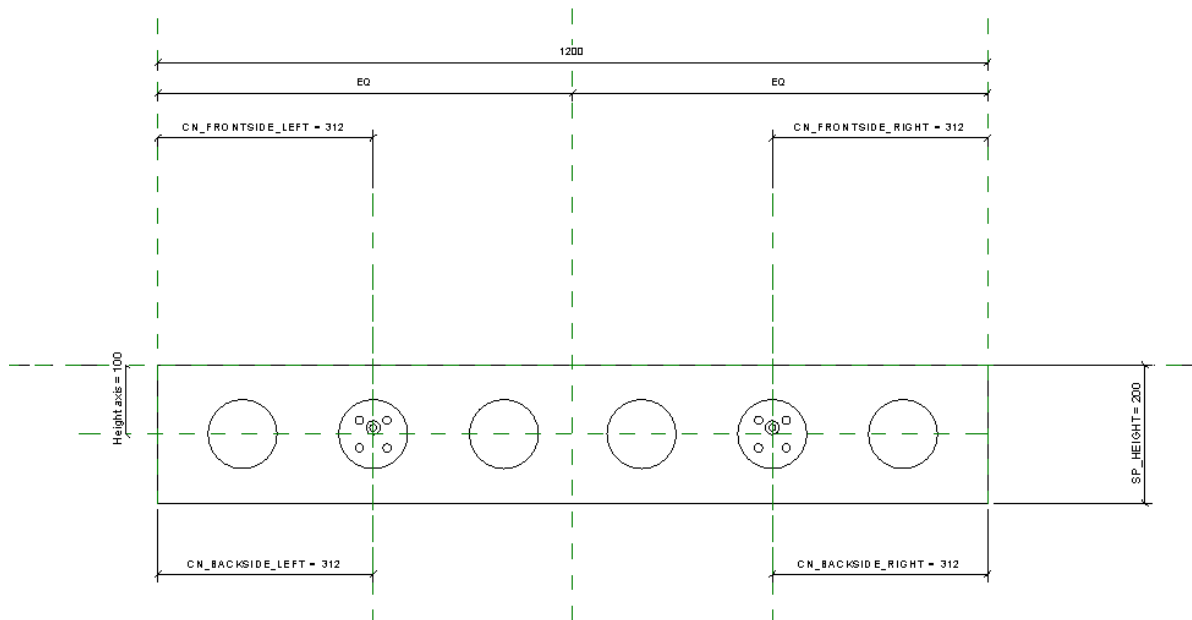


Obrázek 53 - 3D pohled na ztužující stenu DPS

### 3.4.6. Stropní nosník Spiroll

Stropní nosník je konstrukce stropní. Stropy stejně jako stěny jsou rodina systémová. V tomto případě narážíme na stejný problém jako u stěn. U tohoto prvku si však můžeme poradit tak, že stropní desku, tedy Spiroll, který se v systému DPS používá, vytvoříme na šabloně rodiny obecný model liniový a zatřídíme ho do kategorie konstrukční rámová konstrukce. To znamená, že ho budu modelovat jako nosník, a tak ho budu i vykazovat. Jelikož jsou různé druhy Spiroll panelů, které se liší počtem děr, výškou a profilem jak samotného nosníku, tak profilem děr, vytvořil jsem v našem případě jeden vzorový prvek, který použijeme ve vzorovém modelu. Jedná se o panel s výškou 200 mm a 6 dírami, šířka je dána výrobními rozměry linky, tedy 1,2 m.

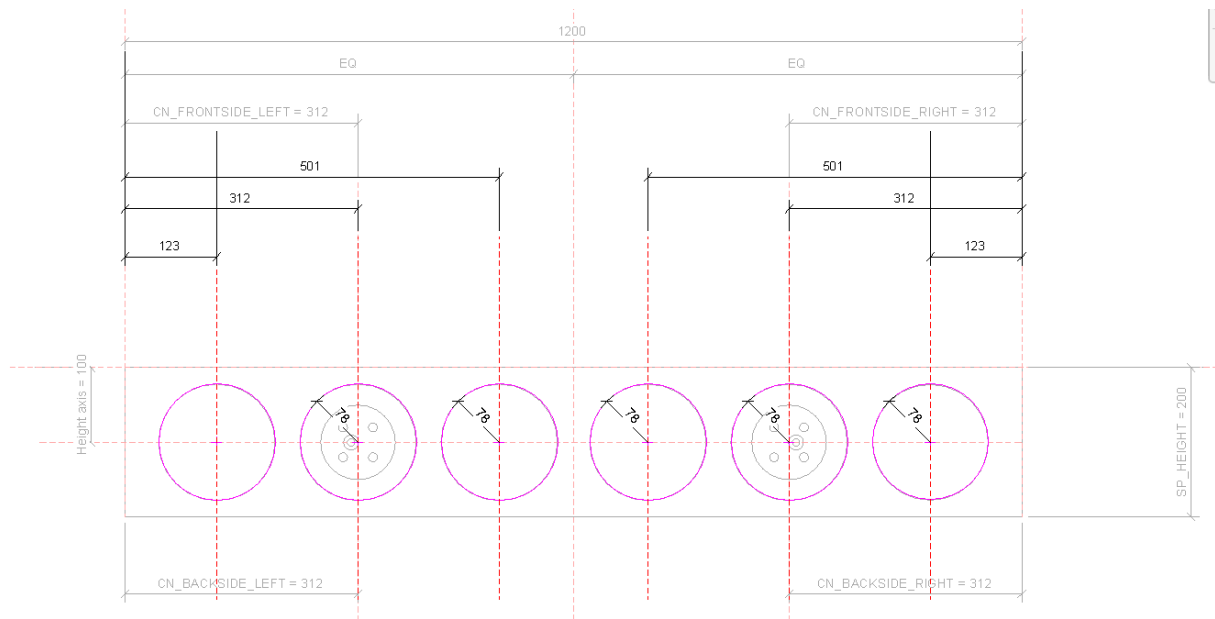
Spiroll je tedy modelovaný jako liniový prvek. Modeluje se tažením křivky, který prochází středem Spirollu při horním okraji. Panel má vložené kotvy, které mají parametrickou vzdálenost od krajů panelu. Toto je z důvodu možnosti použití kotev buď v první nebo druhé dutině a k tomu na každé straně jinak viz obrázek 54.



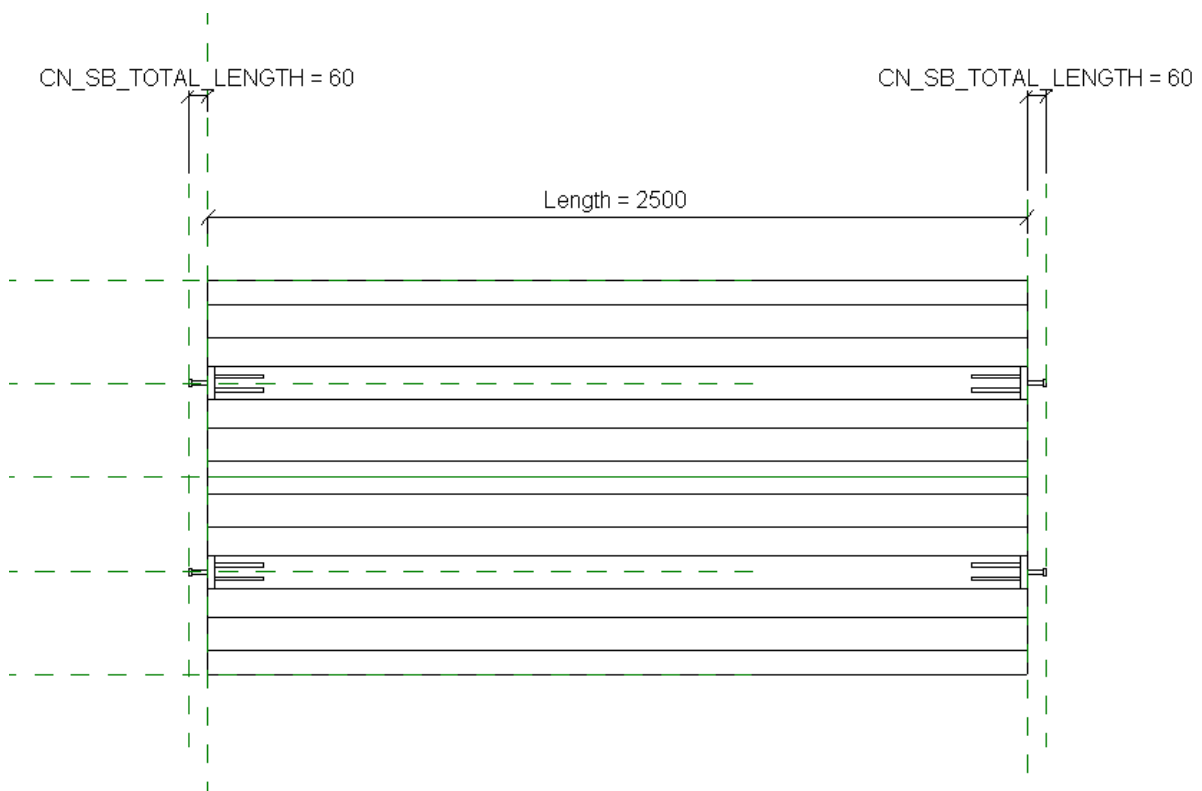
Obrázek 54 - Čelní pohled na rodinu Spiroll panelu

Dutiny jsou modelovány jako dutý tvar a mají napevno daný poloměr, dle vzorového prvku, který jsem převzal z katalogu Spiroll panelů Prefy Brno. Pozice dutin jsou pevně ukotveny kótami viz obrázek 55.

V půdoryse má poté panel přiřazen vestavěný parametr Length jako délka a koncové a počáteční prodloužení, nebo spíše zkrácení v závislosti na celkové délce kotvy Spirollu a nosníku viz obrázek 56.



Obrázek 55 - Parametrizace dutého tvaru ořezání panelu Spiroll

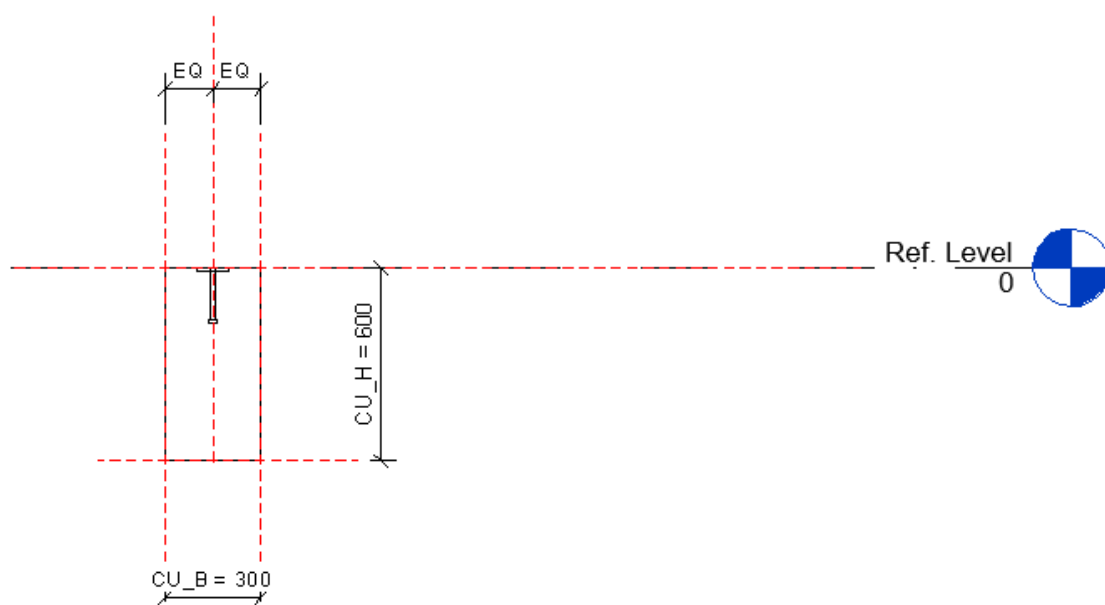


Obrázek 56 - Půdorysný pohled rodiny Spiroll panelu



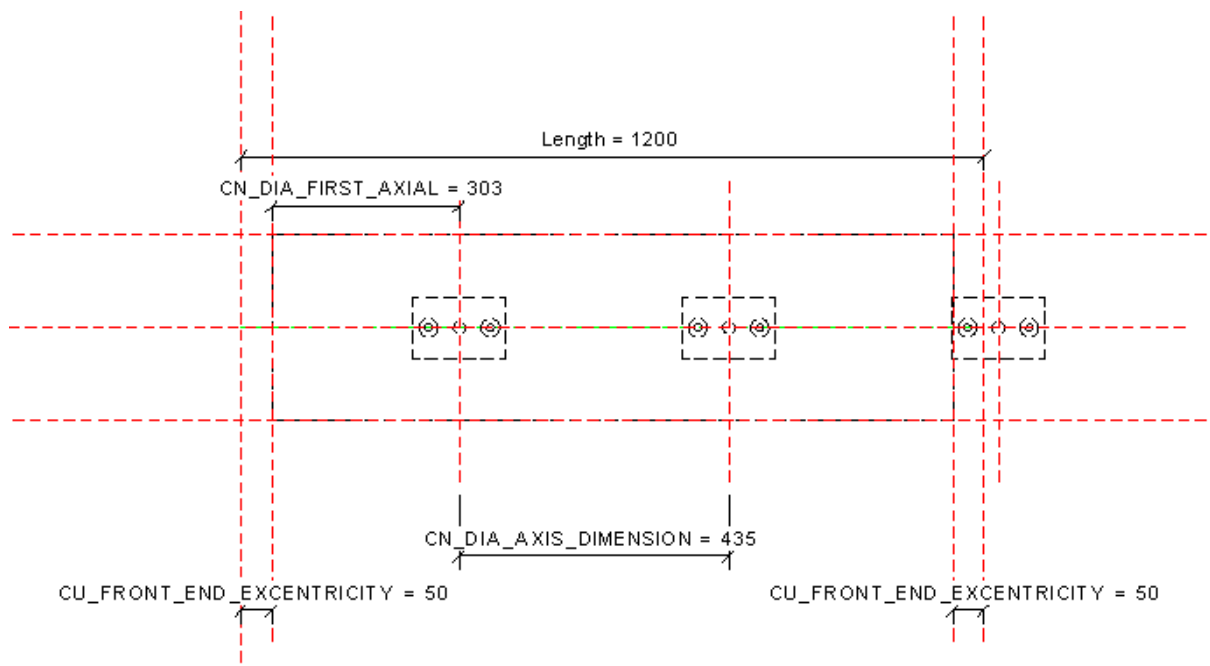
### 3.4.7. Základový práh systému DPS

Základový práh je taktéž rodina založená na šabloně obecný model liniový a následně přiřazena do kategorie konstrukční základové konstrukce. Základový práh má parametrickou výšku a šířku, jak je možno vidět na obrázku 57.



Obrázek 57 - Čelní pohled na rodinu základového prahu

V půdoryse je to u patníku už o něco zajímavější a máme zde už známe kopírování kotev pomocí funkce pole. Funkce pole má poté přiřazený parametr počtu kotev, který se počítá z celkové délky a vzdálenosti mezi kotvami. Kotvy mají mezi sebou nastavený parametr osové vzdálenosti a první řada má přiřazený parametr vzdálenosti od kraje patníku na osu viz obrázek 58. Zároveň má prvek stejně jako ztužující stěna, nastavené parametrické odsazení od kraje, které ovlivňuje celkovou délku.



Obrázek 58 - Půdorysný pohled na rodinu základového patníku

Prvek má odsazení nastavené stejně jako nosník nebo ztužující stěna z toho důvodu, že se modeluje mezi základové patky nebo pasy a je třeba jeho konce odsadit vzhledem k rozměrům přilehlých základových konstrukcí. V tomto případě není sice nutné mít osu modelovanou od bodu k bodu, tedy od osy sloupu k ose sloupu jako je tomu u nosníku abychom zachovali i model analytický ale uživatelsky je to i tak pohodlné.

## 4. Dynamo

### 4.1. Představení Dynama

Dynamo je jednoduché programovací prostředí integrované přímo do softwaru Revit. Jedná se o grafické programování založené na nodech a wirech. Na podobném principu funguje například Grasshopper implementovaný do modelovacího softwaru Rhino. Dynamo se spouští pouze skrze Revit a zároveň lze mít otevřený pouze jeden script na určitý model. Dynamo je od určitého roku součástí Revitu a každý, kdo má zakoupený Revit, má přístup k Dynamu.

Jako nástavba funguje Dynamo tam, kde nástroje Revitu nejsou dostačující. Primárně se Dynamo používá ke zrychlení produkce práce v Revitu, parametrickému modelování, různým analýzám a modelování složitějších tvarů s kterými má Revit obecně problém. Oblastí využití je samozřejmě více a pokud se budeme držet v základním modelovacím prostředí jsou limitovány základními nody. Dynamo je však možné propojit s custom nody, které nesou skript v programovacím jazyce jako Python nebo C#. V tomto případě zase rozšiřujeme využití dynama o další možnosti.

### 4.2. Nody, custom nody a wiry

Jak jsem zmiňoval výše, základní pracovní prostředí nabízí omezený, avšak objemný počet nodů, které můžeme při programování využít. Tam však hranice Dynama vytyčeny nejsou a je možné si nody taktéž vytvářet jako takzvané Custom nodes. Do custom nodů je možné implementovat Python nebo C# skript, který zase rozšiřuje využití a možnosti Dynama.

Jako custom nody se dají stáhnout tzv. packages, tedy knihovny custom nodů vytvořené třetí stranou. Těchto nodů je ve skriptu níže použito spousta a jsou volně ke stažení přímo v prostředí dynama pro každého uživatele. Custom nody od třetích stran mohou být zamčeny, aby nebylo možné nody upravovat nebo popřípadě publikovány někým jiným než autorem. V našem případě jsou ve skriptu použity packages pro dynamo uživatele poměrně základní a všechny bez omezení úprav. Pokud tak dojde k implementaci dynamo skriptu do cizího projektu a uživatel nebude mít stažené packages, které jsou ve skriptu obsaženy, dynamo automaticky nahlásí chybu a nabídne uživateli možnost stažení těchto balíčků. Uživatel pak není limitován žádnými nedostatky nodů v modelu a může skript využívat bez dalších problémů.

### 4.3. Proces propojování, typy vstupů, výstupů, návaznosti

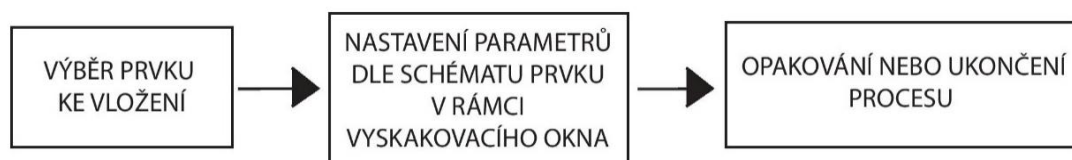
Jednotlivé nody se propojují wiry. Nody mají vstup a výstup definovaný jako například string, double anebo variable. Ne všechny nody mají vstup a výstup jako například nod

categories, kterým definuji kategorii a jako output jí posílám dál. V Dynamu existuje nepsané pravidlo programování zleva doprava, tedy pokud začínáme nový script, vždy jdeme s nody zleva doprava, a tak by měl program i jednotlivé procesy vykonat. Procesy se však dají vždy vůči jiným pozdržet za použití určitého nodu. Tento nod pozdrží operaci, dokud jiná operace nedokončí svůj proces, nehledě na to, zda je před ním nebo za ním. Tento nod si předvedeme přímo ve skriptu, který jsem v rámci diplomové práce vytvořil.

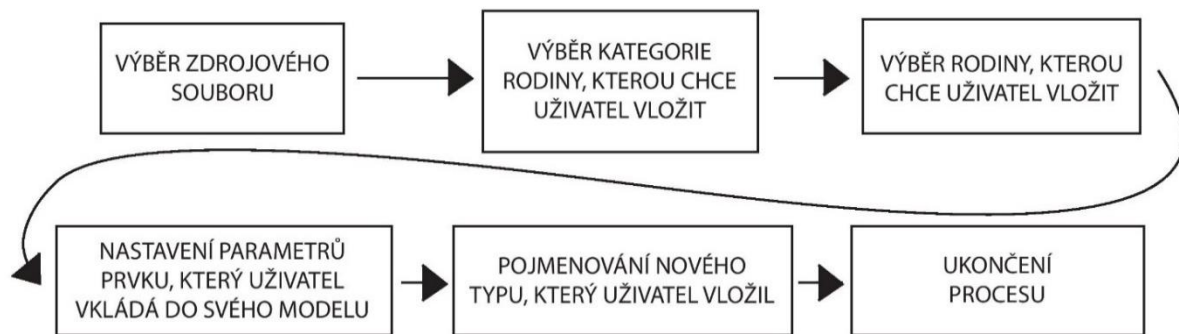
Co se týče vstupů, tak ty mohou být různé. Patří mezi ně integer, variable, double a function. Vstupy musíme při tvorbě skriptu respektovat jinak bude nod hlásit chybu a pravděpodobně nebude fungovat, popřípadě špatně. Samotné nody jsou taktéž specifické v tom, že pokud posíláme do vstupu list, který obsahuje více prvků, nod je schopný list projít prvek po prvku a provést úkon postupně. Nody tedy nativně umí cyklit proces, pokud je jejich inputem list prvků správného charakteru. Toto listování seznamem, který vytvoříme nebo nám vzejde z výběru například všech rodin v projektu skrze nějaký nod, můžeme ovládat pomocí nativního nastavení spojení, které nod nabízí. Spojení můžeme nastavit na automatické, nejdelší, nejkratší a křížení. Pokud tak například budeme mít dva seznamy bodů, z kterých budeme chtít definovat úsečku dle koncových bodů v seznamu 1 a seznamu 2, můžeme tak ovládat jaké body se mezi těmito seznamy spojí. Tedy pokud nastavíme spojení na nejdelší, propojí se úsečkou body o nejdelší vzdálenosti.

#### 4.4. Tvorba dynamo skriptu pro potřeby systému DPS

Původní myšlenka tvorby pluginu do Revitu byla založena na programovacím jazyce C#. Plugin měl sloužit jako proces, kterým by uživatel na základě výběru vložil určitý prvek systému DPS a nastavil by jejich parametry. Schéma původní myšlenky můžeme vidět na obrázku 59. Nicméně při počátečních zkouškách byl zvolen postup tvorby skriptu v programovacím prostředí Dynamo a výsledek se od původní myšlenky trochu liší viz obrázek 60.

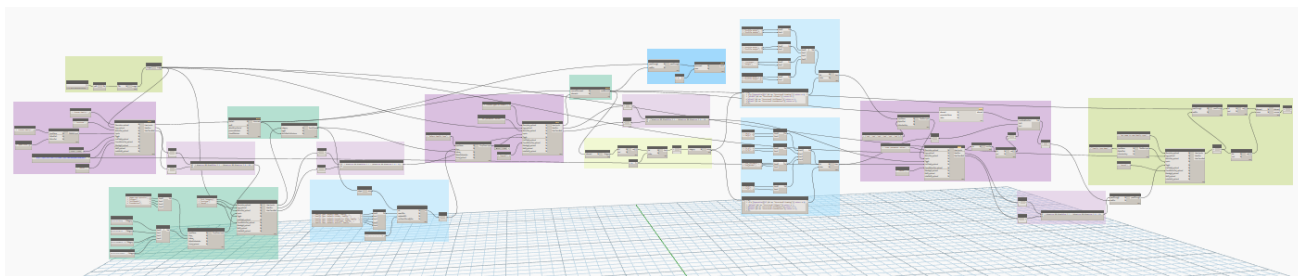


Obrázek 59 - Schéma původního záměru funkčnosti pluginu



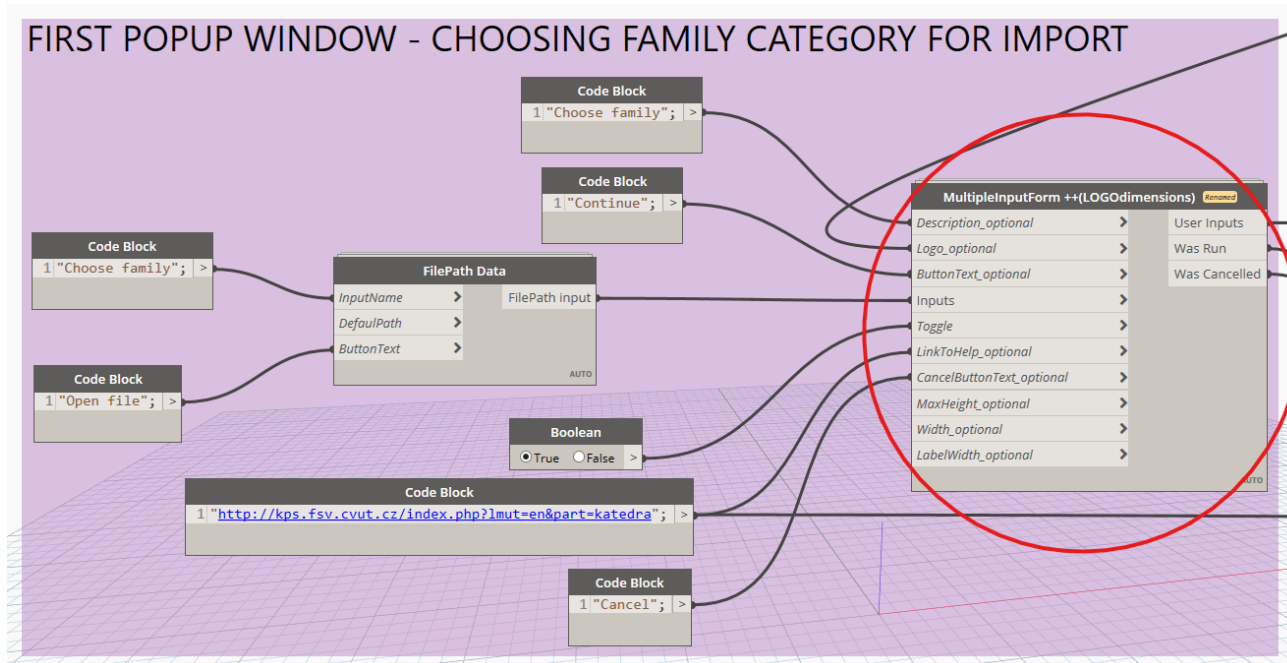
Obrázek 60 - Schéma funkčnosti skriptu

Schéma z obrázku 60 je znázorněno skriptem v dynamu na obrázku 61. Na obrázku vidíme velikost a rozsah skriptu, který byl v rámci diplomové práce zpracován a ve výsledku funguje tak, jak jsem požadoval. Nicméně procesy jsou rozděleny na více částí a vzhledem k možnostem Dynama, které jsem měl možnost vyzkoušet, jsem musel od některých požadovaných vlastností upustit.



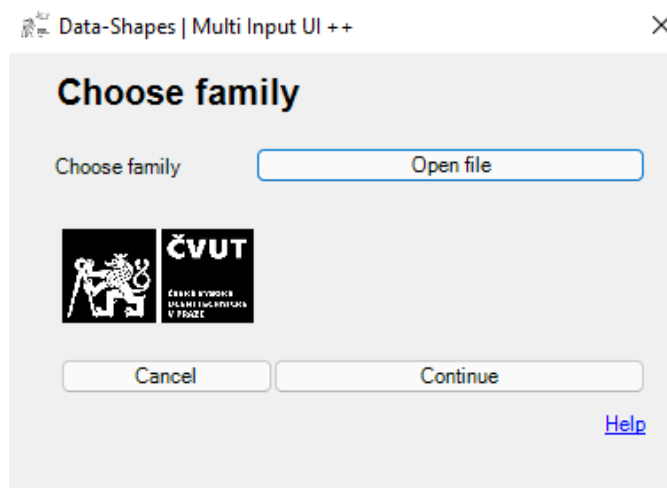
Obrázek 61 - Celkový přehled skriptu pro vkládání prvků DPS

Na obrázku 62 mám první část skriptu pro vyskakovací okno, které uživatel při spuštění procesu uvidí. Vyskakovací okno je vytvořeno nodem, který je na obrázku zakroužkován červeně. Tento custom nod je součástí balíčku od vydavatele Data-shapes. Nod obsahuje python skript se vstupy a výstupy. První část procesu, tedy první vyskakovací okno má vstupy jako název tlačítek nebo webový odkaz. Nejdůležitější je však nod FilePath Data, který umožňuje uživateli otevřít složku a vybrat výchozí soubor viz obrázek č. 61. Mým vzorovým souborem je revit model s prvky systému DPS, z kterého bude skript kopírovat prvky. Jedná se tedy o model, soubor ve formátu .rvt s vloženými rodinami parametrizovaných prvků DPS.



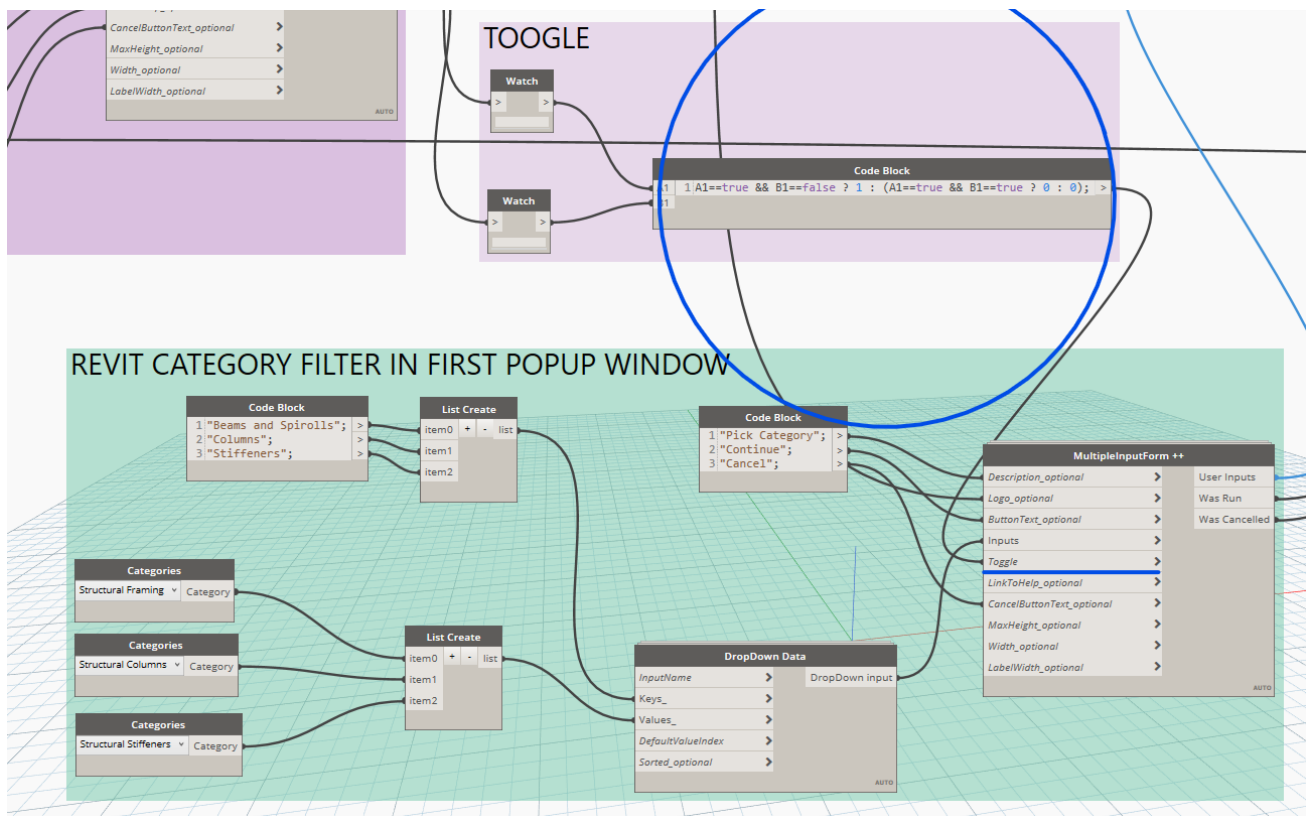
Obrázek 62 - 1. část skriptu systému DPS

Formu vyskakovacího okna můžeme vidět na obrázku 63. Open file jednoduše otevře průzkumníka Windows, uživatel najde v adresáři výchozí soubor a vloží ho do procesu. Pokračuje tlačítkem Continue nebo může ukončit proces tlačítkem Cancel.



Obrázek 63 - 1. vyskakovací okno procesu kopírování prvků

Uživatel má možnost proces ukončit tlačítkem Cancel ve všech fázích procesu, které následují. Avšak samotné tlačítko Cancel, vložené jako input do nodu vyskakovacího okna proces neukončí. A proto je za každým vyskakovacím okénkem proces s code blockem s jednoduchou podmínkou IF viz obrázek 64.



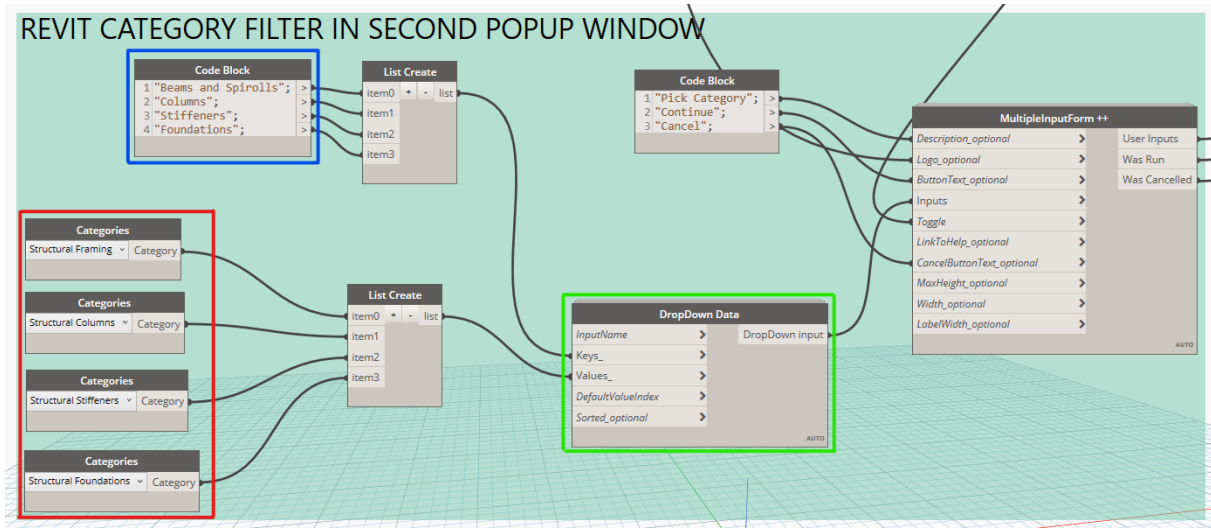
Obrázek 64 - 2. Část skriptu systému DPS

Podmínka IF je závislá na výstupech Was Run a Was Cancelled z nodu MultipleInputForm. Tedy pokud dojde k použití tlačítka Cancel, výstup bude Was Run = true a Was Cancelled = true, tyto dvě hodnoty putují do code blocku pod vstupem A1 a B1. Dle podmínky, pokud máme A1 a B1 = true, bude výstup 0. Výstup následně putuje do vstupu dalšího nodu vyskakovacího okna Toggle, který nodu definuje, zda bude spuštěn nebo ne. Pokud tedy do vstupu Toggle pošleme 0 nebo false, proces nebude spuštěn a dále nebude pokračovat viz obrázek 62. Naprosto opačně to funguje, pokud uživatel zmáčkne tlačítko continue a v procesu má v úmyslu pokračovat. Pokud bude proces pokračovat, výstupem bude Was Run = true a Was Cancelled = false, tedy A1 = true, B1 = false a výstup z code blocku je 1, tedy proces pokračuje.

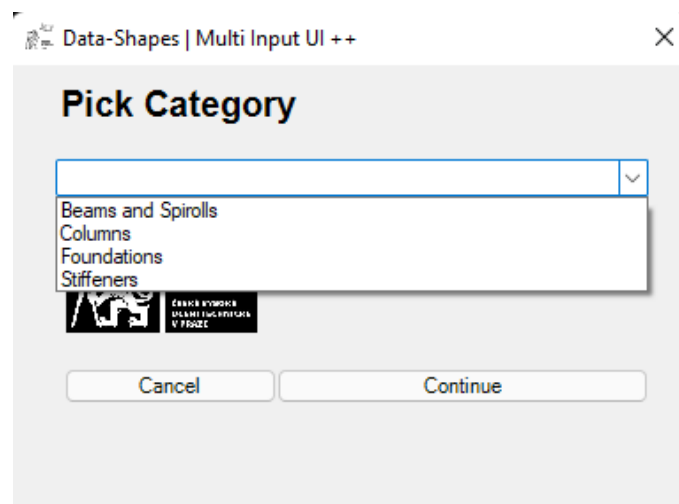
Poté co vybereme v prvním okně výchozí soubor a pokračujeme v procesu, otevře se nám soubor na pozadí Revitu. Další částí procesu je vyskakovací okno s výběrem kategorie rodiny, kterou chceme importovat. Část skriptu, který ovládá další vyskakovací okno můžeme vidět na obrázku 65.

Kategorie jsou do nodu vloženy napřímo a nejsou generovány automaticky. Tomu tak předejdeme neúmyslným implementacím jiných kategorií do procesu, které neobsahují žádné prvky systému DPS. Na výběr máme tak kategorie 4, tedy konstrukční rámovou konstrukci, konstrukční sloupy, konstrukční ztužení a konstrukční základové konstrukce. Výběr kategorie ve vyskakovacím okně funguje přes rozbalovací nabídku. Abych byl schopen

rozbalovací nabídku vytvořit, musí být pro vstup do vyskakovacího okna poslán nod DropDown Data. DropDown Data má potom přes vstup Values napřímo vložen list kategorií a do Keys vložen list názvů, které kategorie přepíše například ze Strctural Framing na Beams and Spirolls. Následně seznam upravených kategorií putuje do vstupu vyskakovacího okna a výběr vypadá následovně viz obrázek 66.



Obrázek 65 - 3. Část skriptu systému DPS

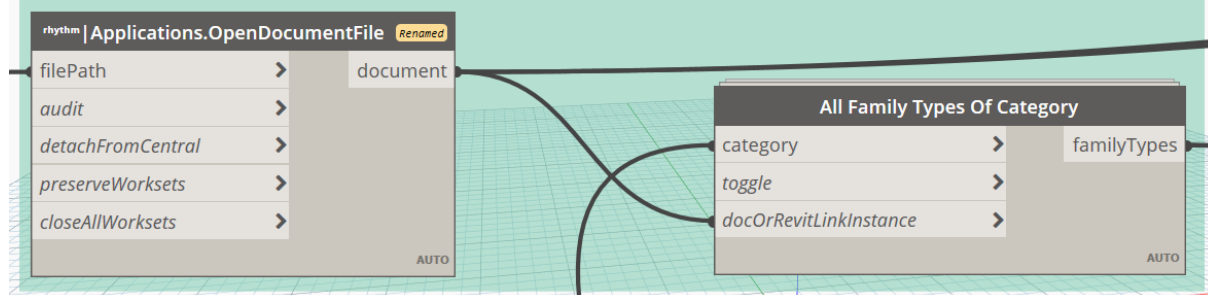


Obrázek 66 - 2. Vyskakovací okno pro výběr kategorie rodiny

Uživatel jednoduše vybere kategorii a pokračuje tlačítkem Continue. V závislosti na vybrané kategorii následuje výstup do série nodů, která z otevřeného dokumentu vybere všechny rodiny dle vybrané kategorie viz obrázek 67. Tedy nod posbírání všech rodin dané kategorie ze zdrojového souboru a v dalším okně nabídne jejich seznam.

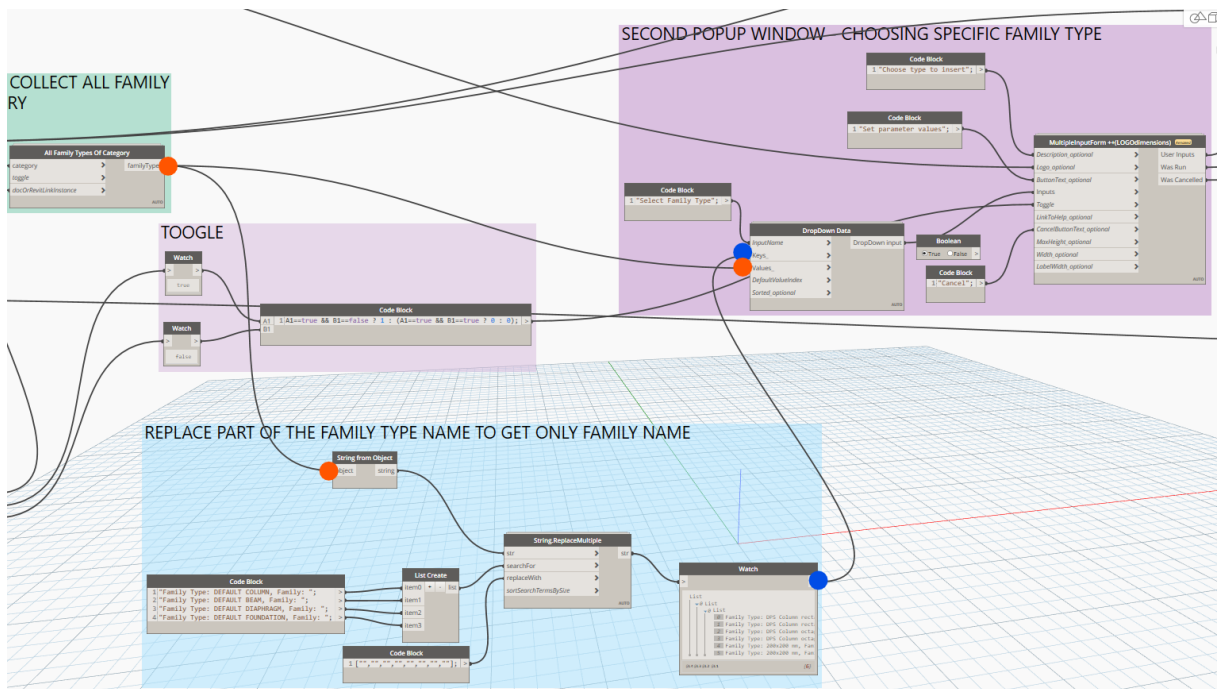


## OPEN RVT. DOCUMENT AND COLLECT ALL FAMILY TYPES OF CHOSEN CATEGORY

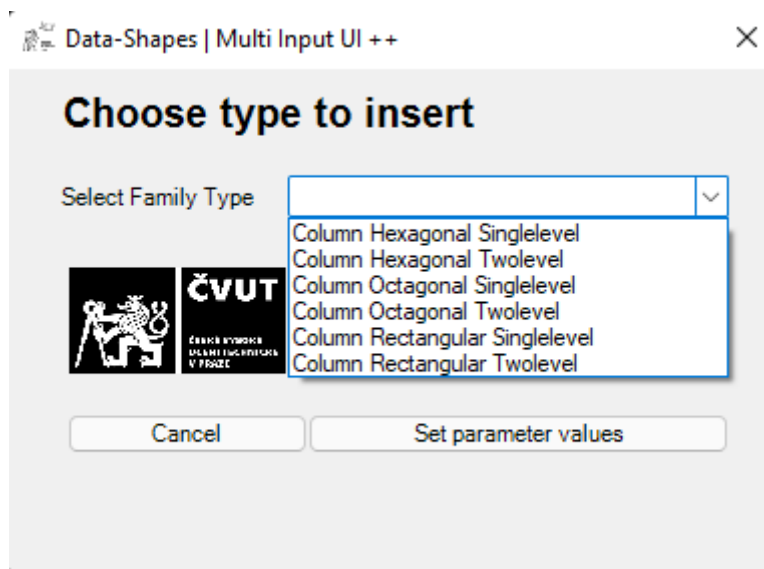


Obrázek 67 - 4. Část skriptu systému DPS

Jeden z výstupů nodu All Family Types Of Category putuje do vstupu nodu DropDown Data Values viz oranžové označení výstupu a vstupu na obrázku 68, abych stejně jako v předchozím vyskakovacím okně definoval výběr pomocí rozbalovací nabídky. Jelikož do hodnot vstupuje rodina a typ, z kategorie námi vybrané, jejich označení není pro uživatele příliš přehledné, tedy se skládá z názvu rodiny a názvu typu. Proto druhý výstup putuje do série nodů, která vezme rodiny a jejich typy a převede je na string, tedy na obecný text. Dále se za pomoci nodu String.ReplaceMultiple nahradí zbytečná část textu neboli označení rodiny za prázdné políčko a zbude pouze název typu. Tento výstup zase v podobě listu, posílám do Keys nodu DropDown Data abych nahradil nepřehledné dlouhé označení za zkrácenou přehlednější verzi viz obrázek 68, modré značení výstupu a vstupu. Vyskakovací okno s výběrem rodiny sloupce můžeme vidět na obrázku 69.



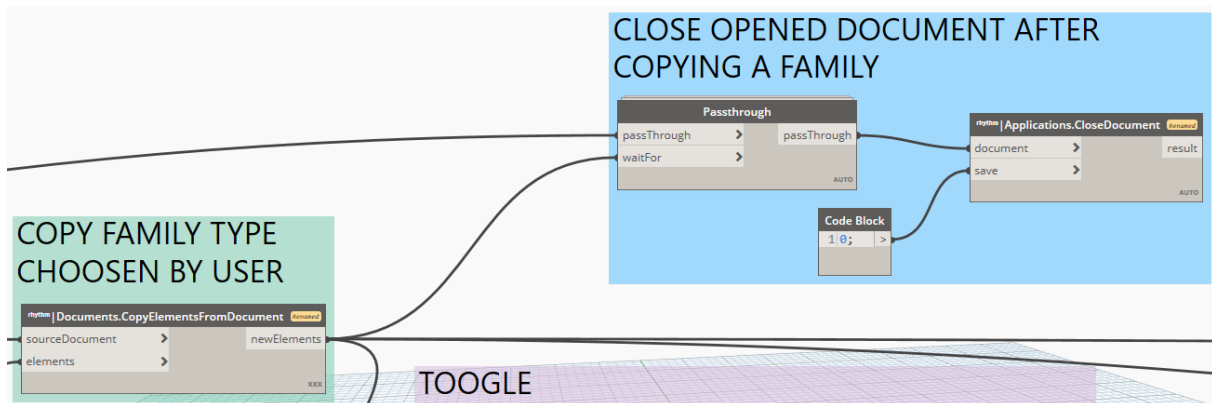
Obrázek 68 - 5. Část skriptu systému DPS



Obrázek 69 - 3. Vyskakovací okno s výběrem rodiny

Uživatel si tak v třetím vyskakovacím okně vybírá rodinu a její typ, který chce vložit do projektu. Nicméně i přesto, že uživatel vkládá rodinu, jeho výběr je založen na zkráceném názvu, jak píšou výše. Tento zkrácený název obsahuje název rodiny. Po této sérii výběru putuje výstup z posledního okna do nodu viz obrázek 70. Nod v zeleném rámečku zkopíruje vybranou rodinu a vloží jí do modelu. Série nodů v modrém rámečku po tomto procesu zavře výchozí dokument, který až do této chvíle běžel na pozadí. Nod Passthrough, který můžeme vidět v modrém rámečku na obrázku 70 používáme i v další části skriptu. Tento nod čeká na

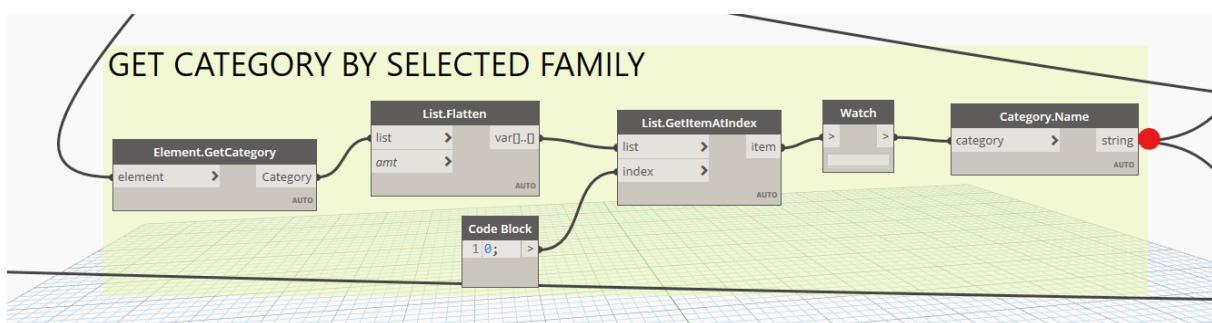
dokončení procesu, vstup waitFor, aby pustil dále jakýkoliv vstup, který nodem prochází, viz vstup passThrough. V tomto případě se tedy soubor nezavře, dokud není dokončen proces výběru rodiny.



Obrázek 70 - 6. Část skriptu systému DPS

Po tomto procesu máme tedy vybranou rodinu vloženou do našeho modelu, ale proces ještě není dokončen. Dalším krokem je nastavení parametrů a pojmenování nově vloženého typu rodiny. Než se dostaneme k další sérii nodu pro další vyskakovací okno, je nutné zmínit tři hlavní procesy, které vedou ke správné funkčnosti.

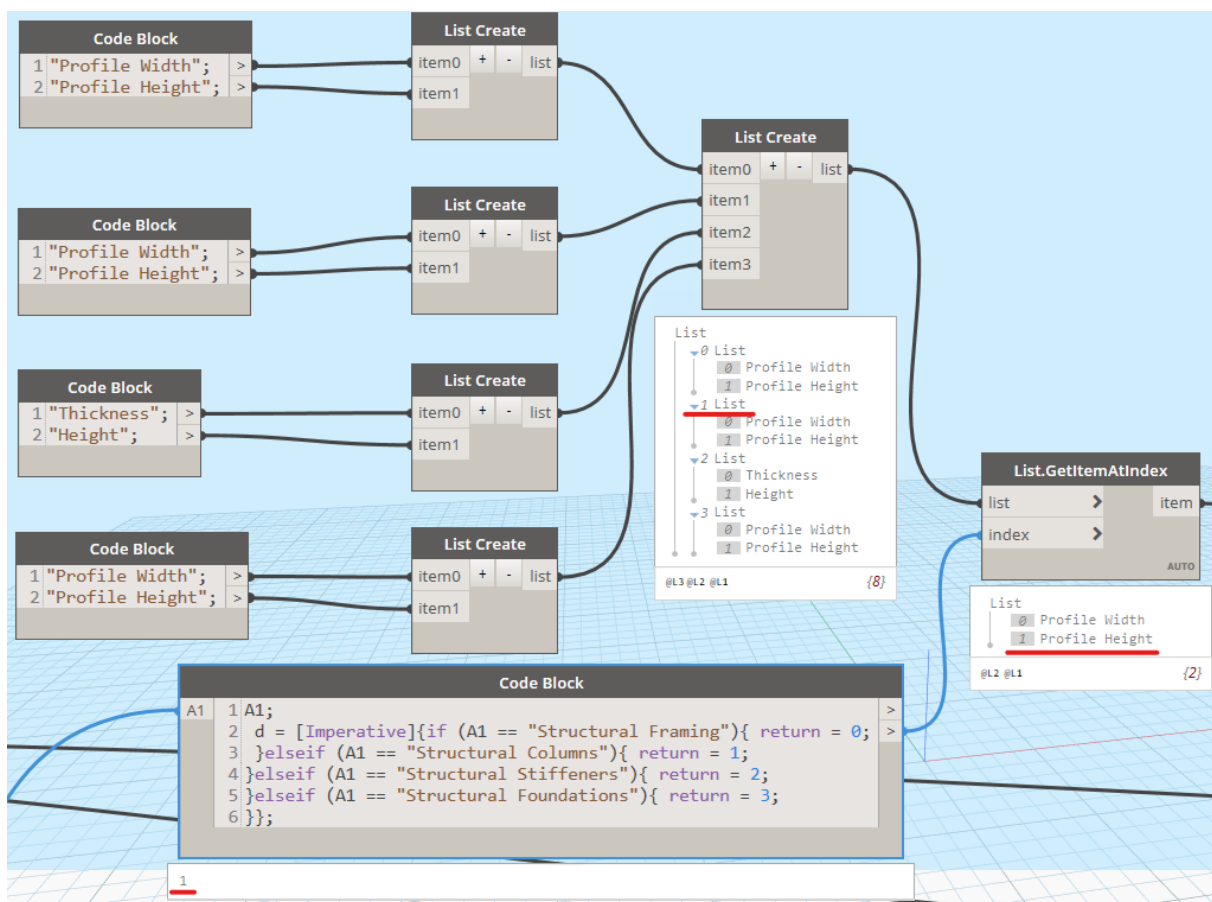
Na obrázku 71 můžeme vidět sérii nodů, které dle vybraného prvku získají jeho kategorii a tu převedou na string neboli text. Tento text dále posíláme do dvou sérii nodů na obrázku 72 a 73.



Obrázek 71 - 7. Část skriptu systému DPS

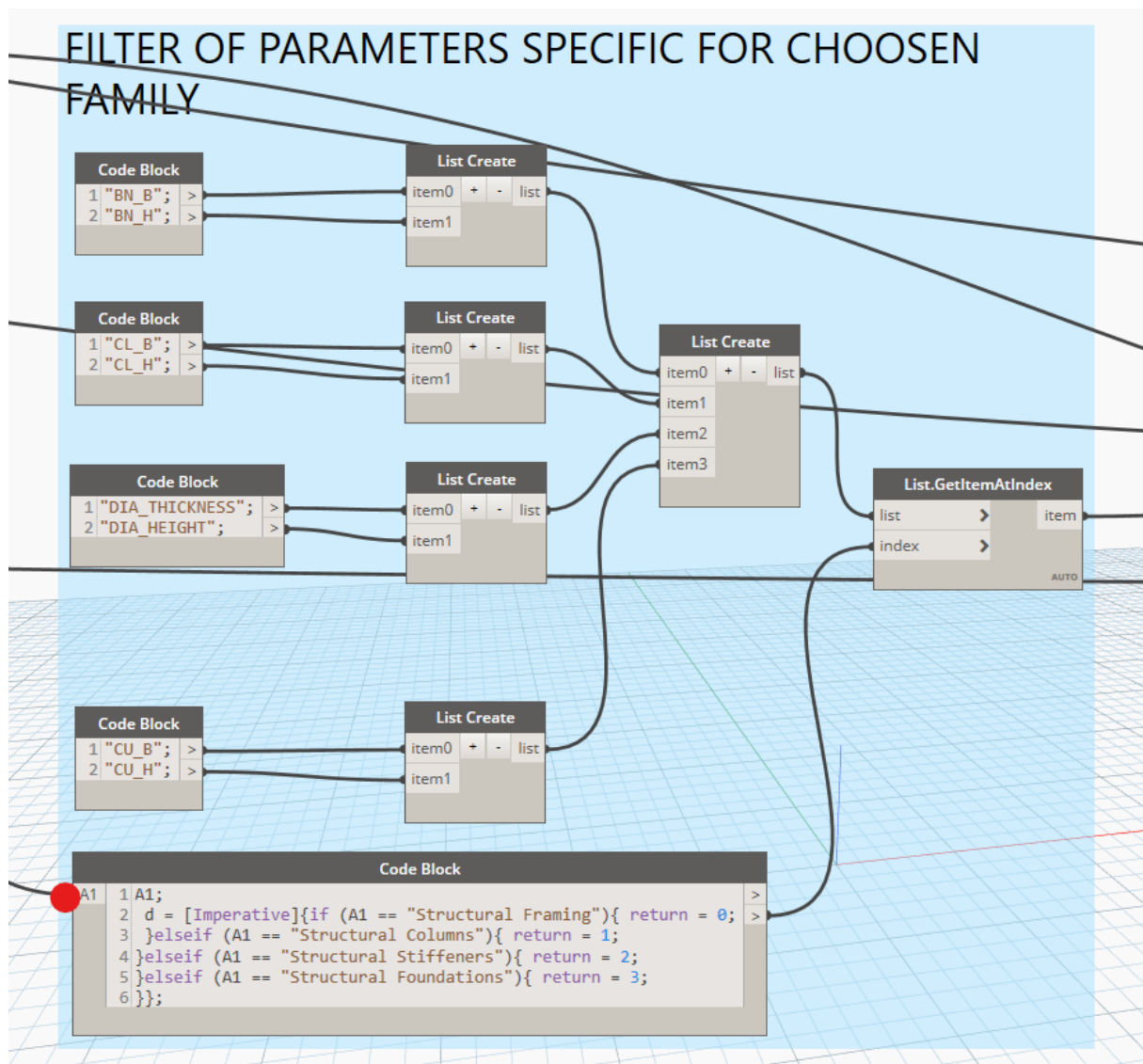
Následující dvě série nodů mohou vypadat podobně ale jejich účel je rozdílný. Nicméně tyto dvě série mají na sebe přímou vazbu. Na prvním obrázku 72 vidíme code block, do které míří právě string z obrázku 71. V codu blocku už máme známou podmínku IF, tentokrát ale v trochu jiném tvaru. Jelikož se dá podmínka IF zapsat několika způsoby, tak

tento je v případě množství obsahu přehlednější. Tedy na základě kategorie rodiny, kterou si uživatel vybere, pošlu string ve tvaru názvu kategorie do code blocku, který na základě tohoto názvu pošle číslo 0, 1, 2 nebo 3. Tyto čísla následně fungují jako index pro node List.GetItemAtIndex, tedy nod, který z listu vybere právě ten prvek pod určeným indexem. Můj list je vytvořen přímo zadanými názvy parametrů nebo jejich přepisem. Pokud jsem tedy zvolil rodinu sloupu, jedná se o kategorii Structural Column a podmínka výstup z podmínky IF bude 1. Tato hodnota putuje jako index do vstupu nodu, který vybere seznam parametrů s indexem 1. Jak vypadá tento seznam můžeme vidět na obrázku 72.



Obrázek 72 – Víceúrovňový seznam

První série vstupů, která tvoří dvouúrovňový list je série názvů typových parametrů, které má v rámci vložení rodiny uživatel upravit. Tento seznam na základě indexu následně pošle do vstupu nodu List.Combine viz obrázek 75.

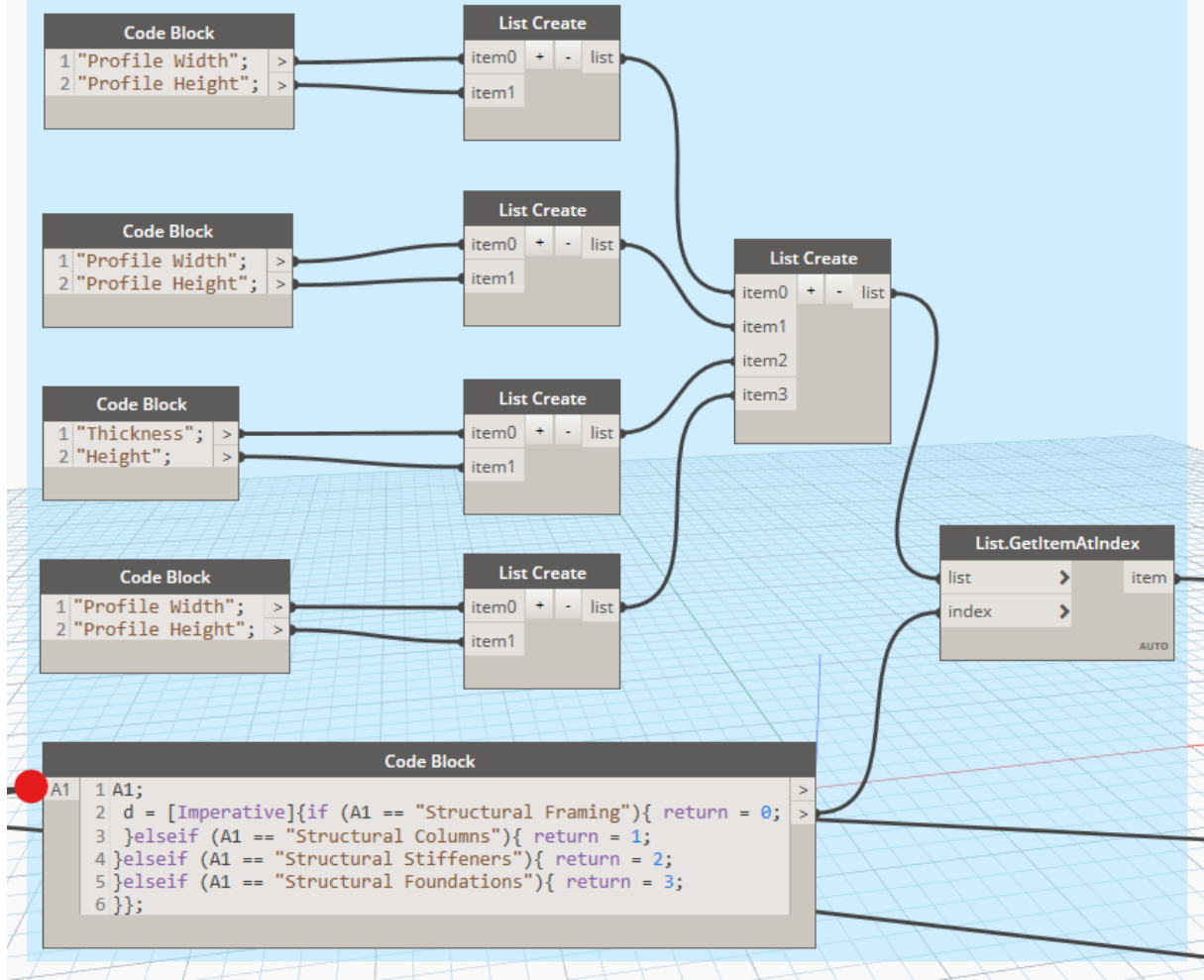


Obrázek 73 - 8. Část skriptu systému DPS

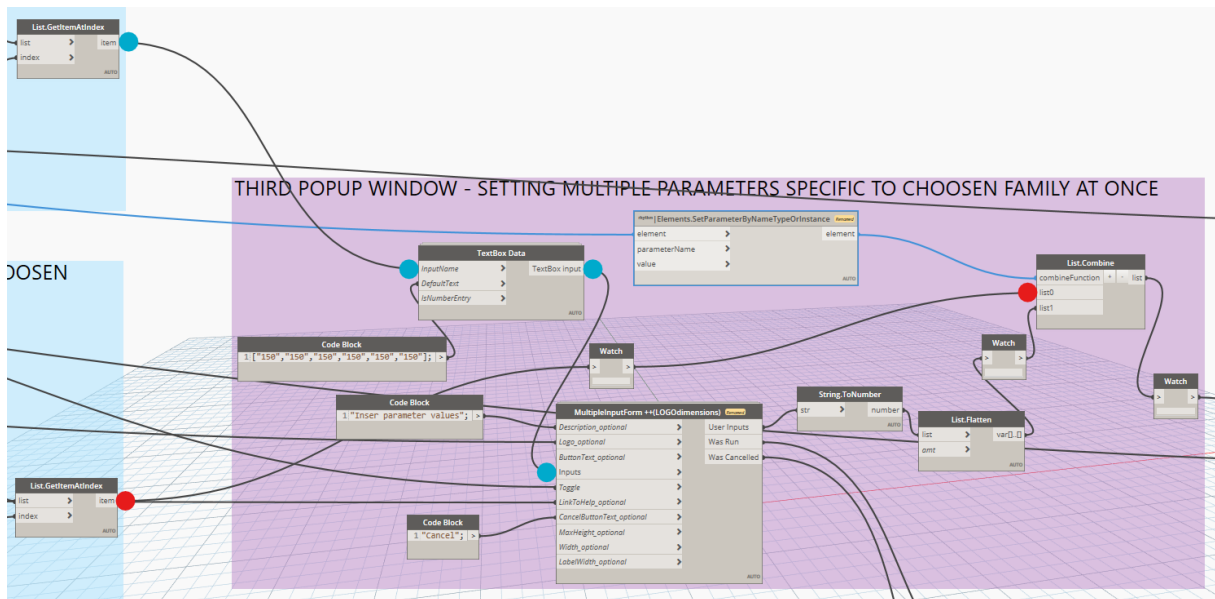
Druhý obrázek 74 vypadá skoro podobně, rozdíl je však pouze ve vstupech do seznamu. Jelikož jsou parametry v prvcích označeny poměrně nepřehledně, jsou v rámci procesu přejmenovány, a právě toto série nodů na obrázku 74 dělá. Výstup v tomto případě vede do vstupu nodu vyskakovacího okna. Jelikož se ale jedná o zadávání hodnot do text boxu, obsah putuje ještě před vstupem do nodu TextBox Data a následně do vstupu nodu vyskakovacího okna.

Výstup je přes nod String.To.Number, který převede text na číslo, následně poslán do nodu List.Combine. Pokud bych poslal do nodu List.Combine vstup ve formě stringu, nebyly by hodnoty parametrům přiřazeny a proces by byl dokončen s chybami.

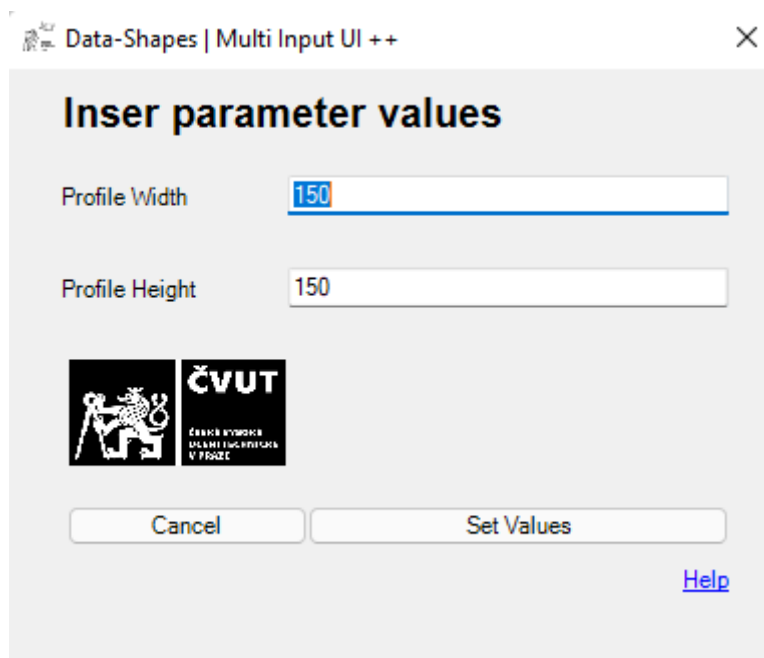
# RENAME OF SPECIFIC PARAMETERS



Obrázek 74 - 9. Část skriptu systému DPS



Obrázek 75 - 10. Část skriptu systému DPS



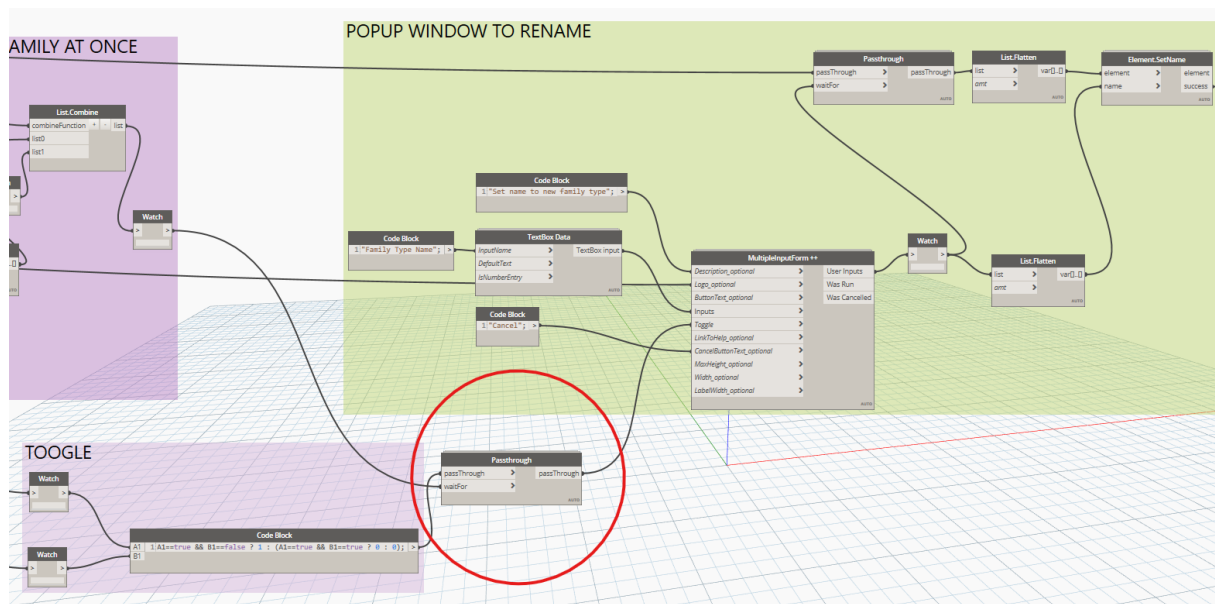
Obrázek 76 - 4. Vyskakovací okno pro zadání hodnot parametrům vložené rodiny

Nod List.Combine má tedy 2 vstupy. Seznam parametrů a seznam hodnot, které uživatel vložil ve vyskakovacím okně. Třetí vstup nebo výstup vede do nodu Element.SetParameterByNameTypeOrInstance. Tento parametr následně přiřadí hodnotu uživatelem zadané hodnoty referenčnímu parametru. Vyskakovací okno s možností zadání parametrů vidíme na obrázku 76.

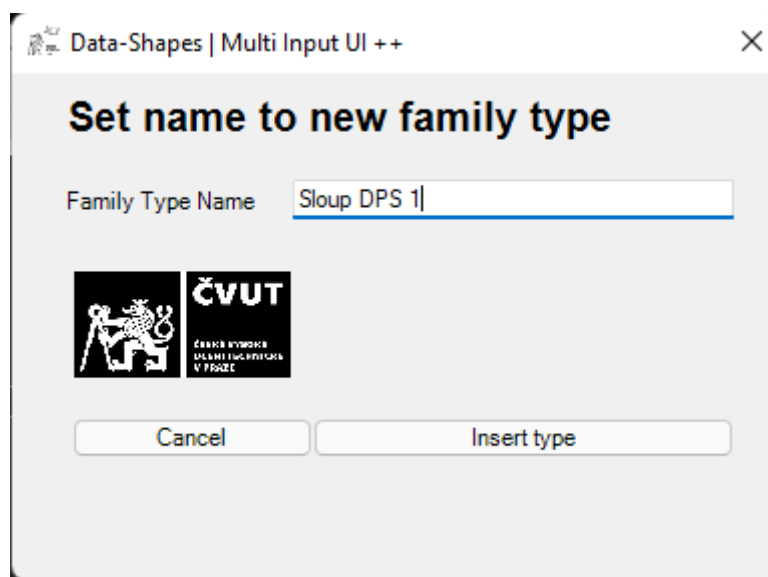
V poslední části musí uživatel zadat jméno nového typu, který už byl do výchozího projektu vložen. Následující série nodů tedy upravuje název typu rodiny, která už byla vložena. Původní myšlenka byla, že uživatel zadá název typu v rámci vyskakovacího okna při zadávání parametrů, nicméně to nebylo tak jednoduché, jelikož název typu není parametr jako například parametr šířky nosníku. Pokud chci přejmenovat typ, je nutné použít nod Element.SetName viz obrázek 77 a ten se mi do předchozího procesu nepodařilo zdárně implementovat.

Poslední vyskakovací okno má referenční sérii nodů na obrázku 77. Předtím, než se proces spustí, čeká na ukončení procesu úpravy parametrů viz červený obláček na obrázku. Následně je zde klasický nod vyskakovacího okna se vstupem do text boxu a výstup, tedy uživatelsky zadaný název do nodu Element.SetName, nod který změní jméno nově vytvořeného typu.

Poslední vyskakovací okno procesu tak můžeme vidět na obrázku 78. Po dokončení zkontrolujeme vložení v modelu, zda se typ vložil pod správným jménem a správně zadanými parametry. Můj případ, který jsem si v procesu vytvořil můžeme vidět na obrázku 79.



Obrázek 77 - 11. Část skriptu systému DPS



Obrázek 78 - 5. Vyskakovací okno zadání názvu nového typu



Type Properties



Family:

Type:

Type Parameters

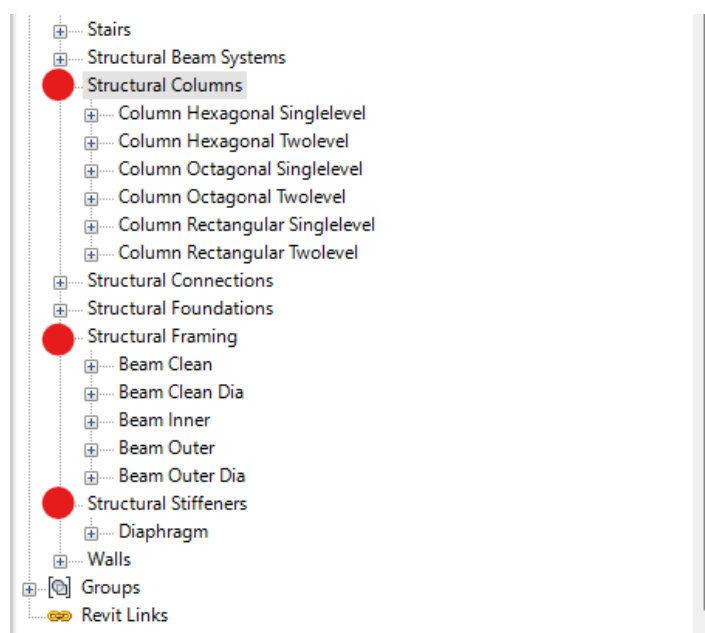
Parameter	Value	
<b>Text</b> ^		
TYPE_MARK		
<b>Structural</b> ^		
Section Shape	Not Defined	
<b>Dimensions</b> ^		
CL_B	150,0	
CL_H	150,0	
<b>Identity Data</b> ^		
Assembly Code		
Cost		
Description		
Keynote		
Manufacturer		
Model		
Section Name Key		
<b>Type Comments</b>		
Type Image		
URL		
Assembly Description		
<b>Type Mark</b>		
OmniClass Number	23.25.30.11.14.11	
OmniClass Title	Columns	
Code Name		

[What do these properties do?](#)

Obrázek 79 - Vložený prvek vygenerovaný skriptem

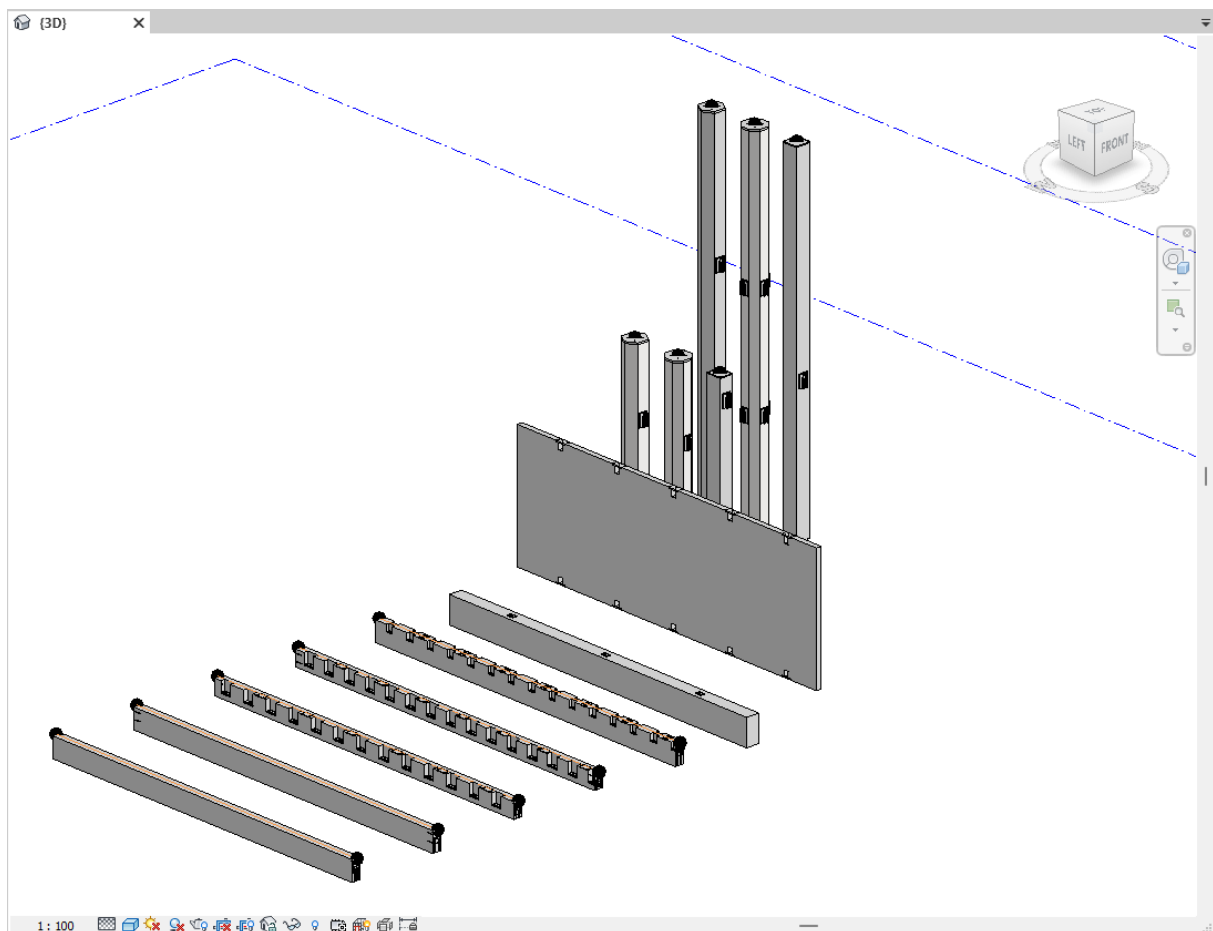
## 4.5. Implementace skriptu do projektu a ukázka funkčnosti

Jak bylo zmíněno výše, na začátku popisu skriptu, v procesu si uživatel vybírá zdrojový soubor, tedy knihovnu prvků. Takto knihovna, neboli soubor ve formátu .rvt, obsahuje všechny prvky systému DPS viz obrázek 80. Ostatní rodiny a nepoužité prvky jsou ze zdrojového modelu odstraněny příkazem `purge unused`, tedy odstranit nepoužité. Všechny prvky v 3D pohledu můžeme vidět na obrázku 81.

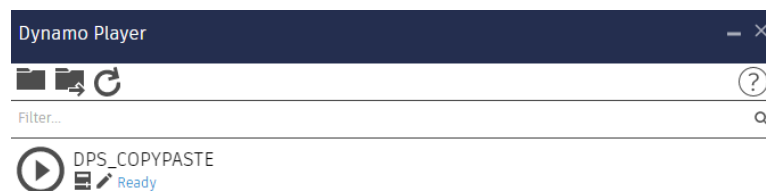


Obrázek 80 - Dostupné rodiny ve zdrojovém souboru

Pokud bude chtít uživatel spustit skript nemusí přitom složitě otevírat Dynamo a nahlížet do skriptu, který by mohl napříkladem i nechtěným zásahem změnit, a tak způsobit jeho nefunkčnost ale může využít Dynamo playeru, který je součástí Revitu. Dynamo player slouží, jak je naznačeno v názvu k přehrání scriptů vytvořených v Dynamu. Player zobrazuje vždy skripty z jedné výchozí složky, kterou si uživatel nastaví. Pokud by tedy docházelo k implementaci do BIM workflow třetích stran je nutné na toto myslet a skript uložit na správné místo.



Obrázek 81 - 3D zobrazení dostupných rodnin ve zdrojovém souboru



DPS Source File.rvt

Obrázek 82 - Dynamo player

Pokud uživatel zmáčkne tlačítko play, skript se spustí a projde procesem výběru souboru, určení kategorie, rodiny, prvků a názvu, jak je znázorněno v rámci popisu skriptu. Původní myšlenka, kterou jsem chtěl aplikovat, bylo proces opakovat, pokud by měl uživatel zájem. Tedy jednoduchým ano/ne by určil, zda se má proces znovu spustit, například od místa výběru kategorie. Nicméně Dynamo jako takové není na cyklení procesů příliš přizpůsobeno. I na samotnou tvorbu takového skriptu, není úplně ideální volba Dynamo a pokud se tvoří takové pluginy, nejčastěji se používá programovací jazyk C# nebo Python. Jak můžeme vidět, použití Dynama není úplně vyloučeno a mně se většina cílů podařila vytvořit. Nicméně některé procesy se musely zbytečně rozdělit a místo původní myšlenky pár vyskakovacích oken, kde by uživatel zadal vše potřebné jsem musel proces rozdělit na celkem oken 5. Pokud se tedy vrátíme k cyklení, samotný skript tuto funkci nemá ale vlastně stačí, aby uživatel znovu zmáčknul tlačítko přehrát a může procesem projít znovu. Nicméně bude muset znovu vybírat zdrojový soubor, který je vždy stejný.

## 5. Model z dynamických prvků systému DPS

V rámci ukázky funkčnosti dynamických prvků v akci jsem vytvořil dva referenční modely. Model č. 1 je vytvořen pouze z prvků systému DPS, a jedná se tak o model nosných prvků. Model č. 2 je vytvořen z části prvků z modelu č. 1 a je doplněn o další nezbytné prvky ze systému DPS, které zajišťují funkčnost a stabilitu objektu. Na modelu č. 1 následně předvedu, co se můžeme s BIM modelem provádět, jako je převod do statického softwaru nebo tvorba harmonogramu a procesu výstavby. Model č. 2 je vytvořen z toho důvodu, abych představil možnost recyklace prvků systému DPS, formou jejich znovupoužití na jiné stavbě s vazbou na BIM model. Následně si předvedeme tvorbu dokumentace a výkazy, které jsme schopni díky parametrizaci systému vytvářet.

### 5.1. Tvorba modelu z prvků DPS

Model tvořený z prvků DPS, tedy převážně sloupů, nosníku a základových konstrukcí je poměrně snadný a rychlý. Revit uživateli umožňuje modelování těchto prvků jednoduše spojit s předem připravenou osnovou.

V rámci diplomové práce jsem vytvořil dva rozdílné typy budov, pro ukázkou tvorby skeletového systému z prvků DPS. Tyto dva modely, jsou vytvořeny skoro ze stejného množství i stejných typů prvků, abychom zároveň poukázali na možnost demontáže a znovu použití systému na rozdílném typu budovy. Ukázka je provedena na jednoduchém a poměrně malém dvoupodlažním objektu, výchozím modelu, který může být využit například jako menší montážní hala s přilehlými prostory kanceláří a zázemí montáže. Druhá budova, kterou stavím více do výšky by poté mohla fungovat například jako radnice nebo malá kancelářská budova.

Parametrické prvky, které jsou použity, mají značení, které rozdílně označuje prvky z modelu č. 1 a v modelu č. 2. Pokud tedy budou prvky z modelu č. 1 použity v modelu č. 2, budou mít značení z modelu 1 a můžu tak ve výkazech tímto označením prvky rozlišit. Taktéž bude možné prvky porovnat z hlediska počtu a procentuálně ukázat výhody prefabrikovaného systému, ve smyslu znovu použití prvků, které mohou mít životnost reálně větší než 50 let.

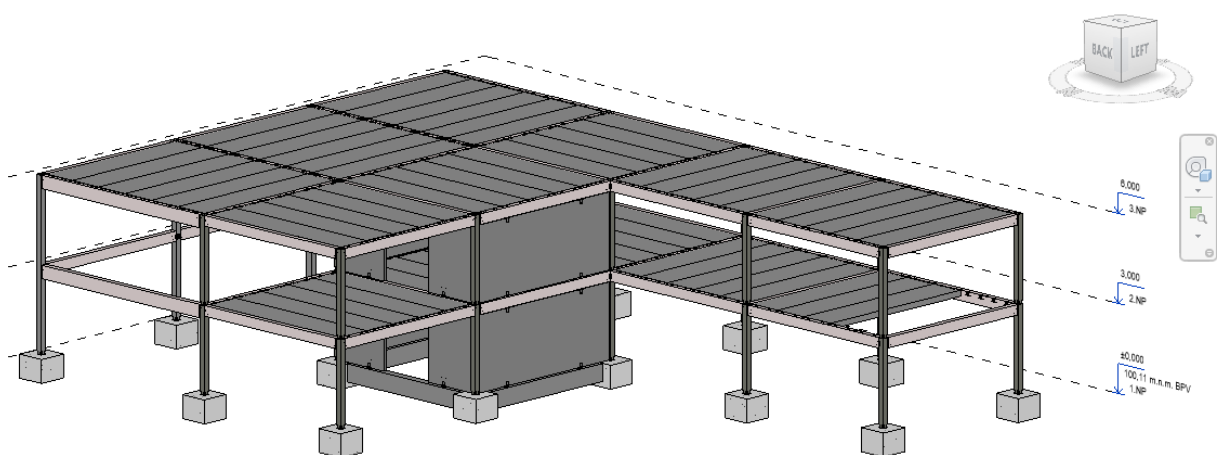
Prvky, jak je psáno výše jsou značeny dle kódu sestavy, označení typového a označení prvkového. Toto značení je upraveno vzhledem k modelu č. 1 a 2. Tedy první dva prvky jako kód sestavy a označení typové jsou stejné a označení instanční je poté ve formátu MA01 až MAxx pro model č. 1 a MB01 až MBxx pro model č. 2. Příklad značení při sloučení označení ve výkazu můžeme vidět na obrázku 83.

# TM01.01.MA01

KÓD TYPOVÉ INSTANČNÍ  
SESTAVY OZNAČENÍ OZNAČENÍ

Obrázek 83 - Značení prvků DPS ve vzorových modelech

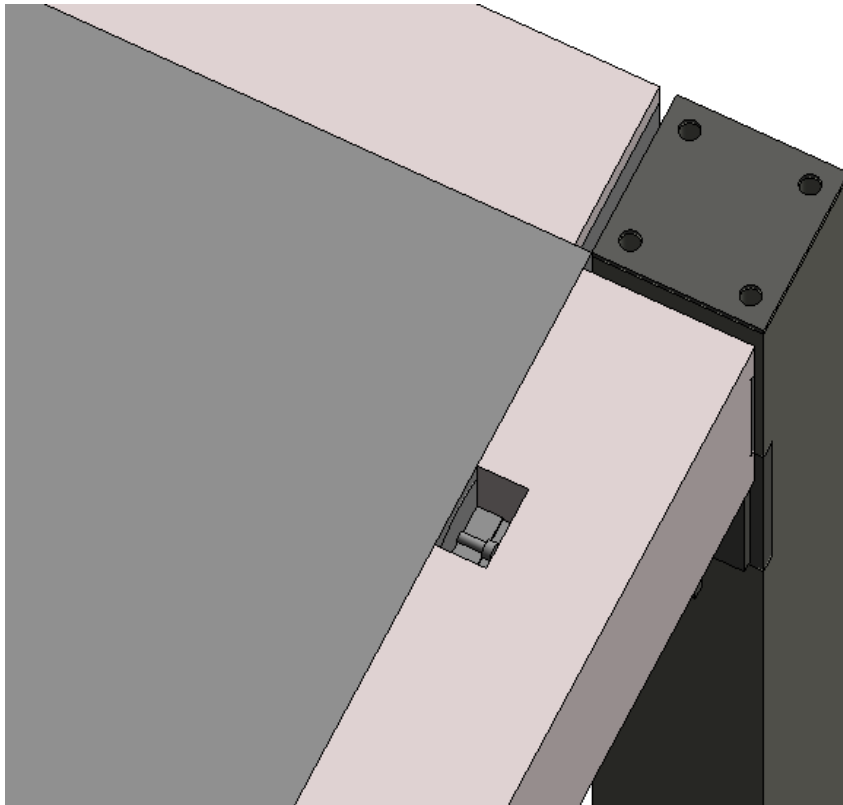
První model č. 1 můžeme vidět na obrázku 84. Model je složen pouze z konstrukčních prvků a základových konstrukcí. Jedná se o sloupový systém ztužený výztužnými stěnami.



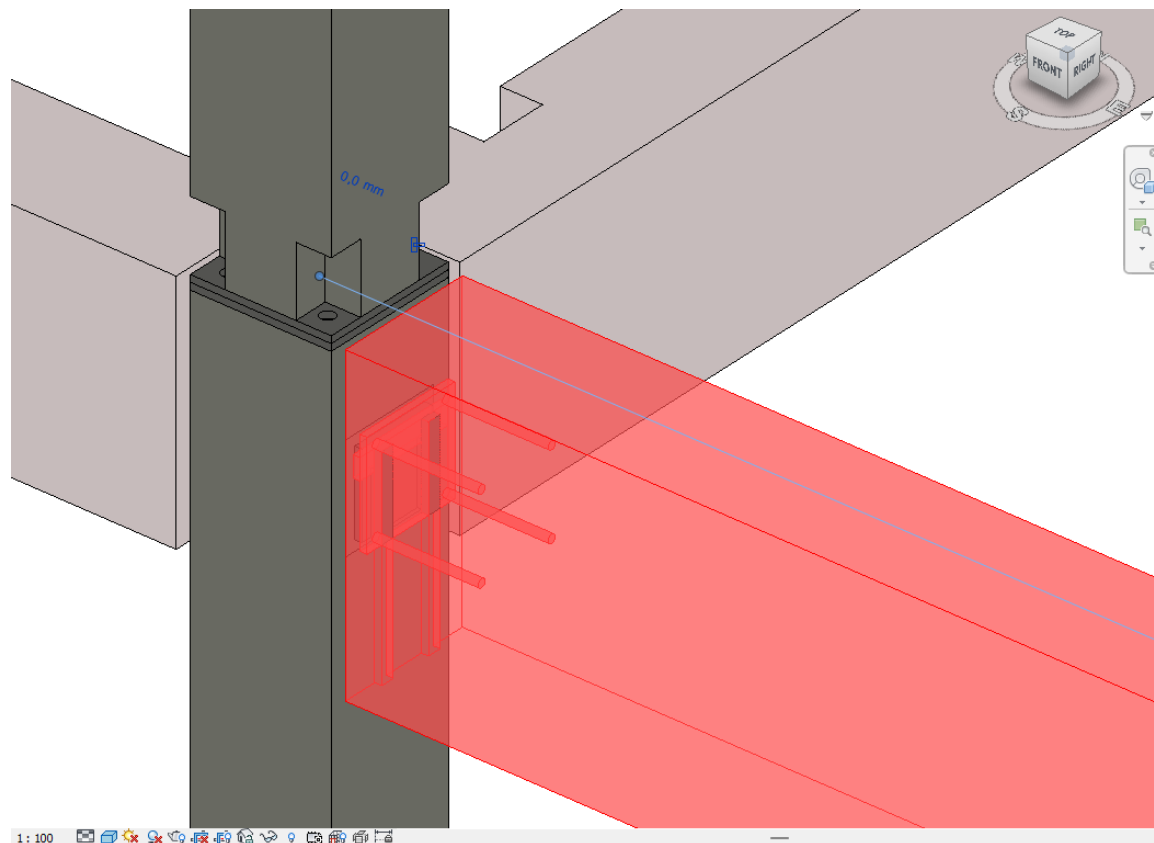
Obrázek 84 - 3D pohled na model č.1

Dynamické rodiny konstrukčních prvků mezi sebou fungují stejně jako v realitě. Kotvy mají reálné rozměry a zapadají do sebe tak, jak by měly. Příklad spoje nosník sloup nosník, můžeme vidět obrázku 85 a 86.

Jelikož jsem celou dobu mluvil i modelu analytickém, který nás po celou dobu tvorby rodin omezoval, předvedeme si taktéž jeho ukázkou na referenčním modelu a přenos modelu do softwaru Scia engineer a Robot Structural, kde spustím jednoduchý výpočet vnitřních sil od vlastní tíhy, abych zkontroloval přenos sil ve styčnicích. Přenos si předvedeme i přes formát IFC, na jednoduchém rámu a následně napřímo z Revitu do Robotu na kompletním modelu.



Obrázek 85 - Detail vazby prvků systému DPS



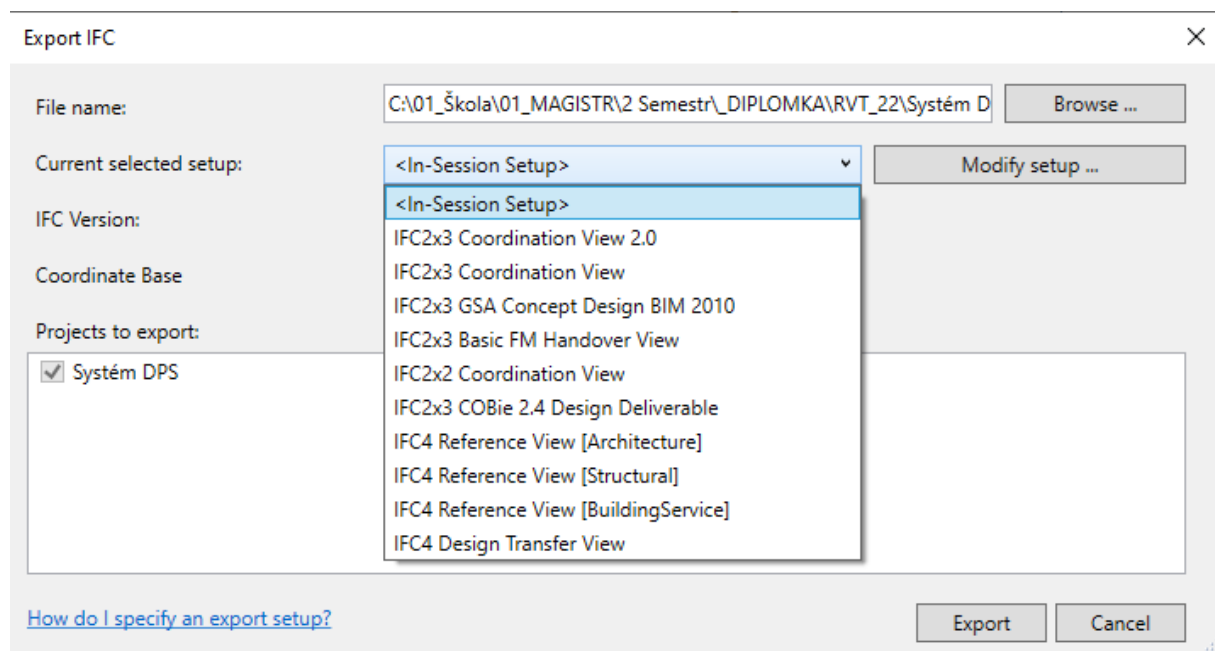
Obrázek 86 - Detail vazby prvků systému DPS 2

## 5.2. Převod analytického modelu ve formát IFC

Zkráceně IFC neboli Industry Foundation Classes je formát pro přenos dat grafických, informačních nebo materiálových mezi softwary pracující s BIMem. Jedná se o formát, který se řídí evropskou normou EN ISO 167398 neboli původně standard IFC4. O formát IFC se stará skupina z Evropského výboru pro normalizaci zabývající se oblastí BIMu.

IFC jako formát, musí mít jasně stanovený účel. Výměna dat přes tento formát nemusí totiž vždy fungovat naprosto správně. Prvky sice mezi softwary převedete, budou mít správné typové zařazení ale jejich úprava nemusí být možná jako například převod geometrie z ArchiCadu do Revitu. Z tohoto důvodu se zpracovává MVD, Model View Definition, tedy požadavky na zobrazení a množství informací modelu předávaného IFC formátem.

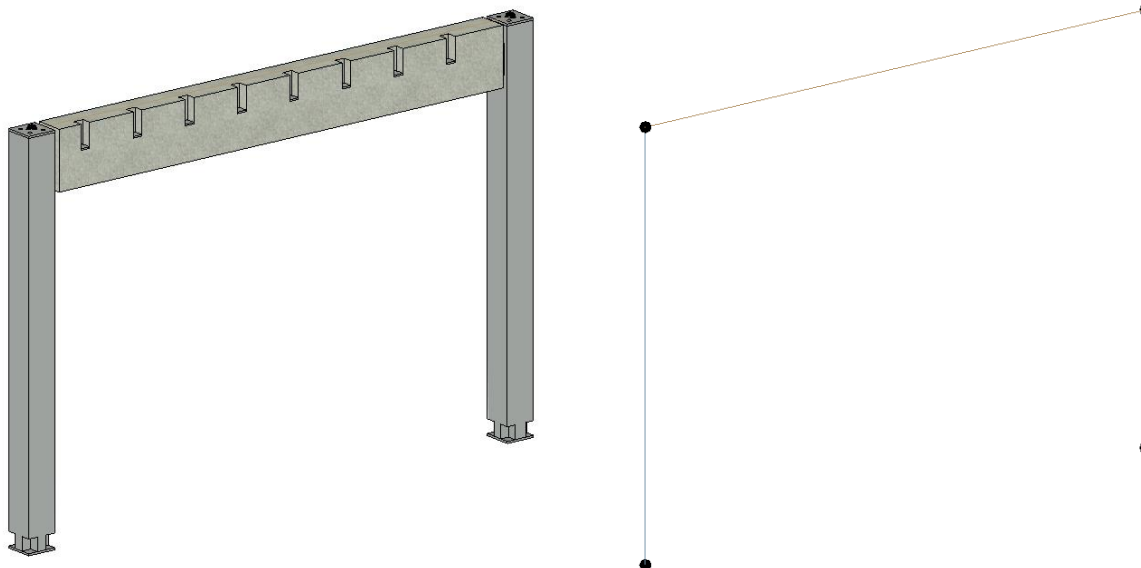
V mém případě by šlo primárně o přenos analytického modelu, tedy síťového 3D modelu a geometrii prvků. Formát IFC má jako takový různé formy označované nejčastěji 2x2 nebo 2x3 s dovětkem. Nejaktuálnější forma je 2x3 Coordination View 2.0, forma 2x2 je starší a potom máme například 2x3GSA Concept Design BIM 2010 formu, která zpracovává data pro US Government Services Administration, něco jako státní americké úřady. Pokud exportujeme model do IFC formátu z Revitu, je možné, si při exportu nastavit právě určitý formát viz obrázek 87.



Obrázek 87 - Export modelu rámu do IFC



Abychom si ukázali, jak je možné formát IFC pro analytický model využít, je znázorněn na jednoduchém příkladu příčné vazby viz obrázek 88. Pro převod použijeme formát 2x3 Coordination View 2.0.



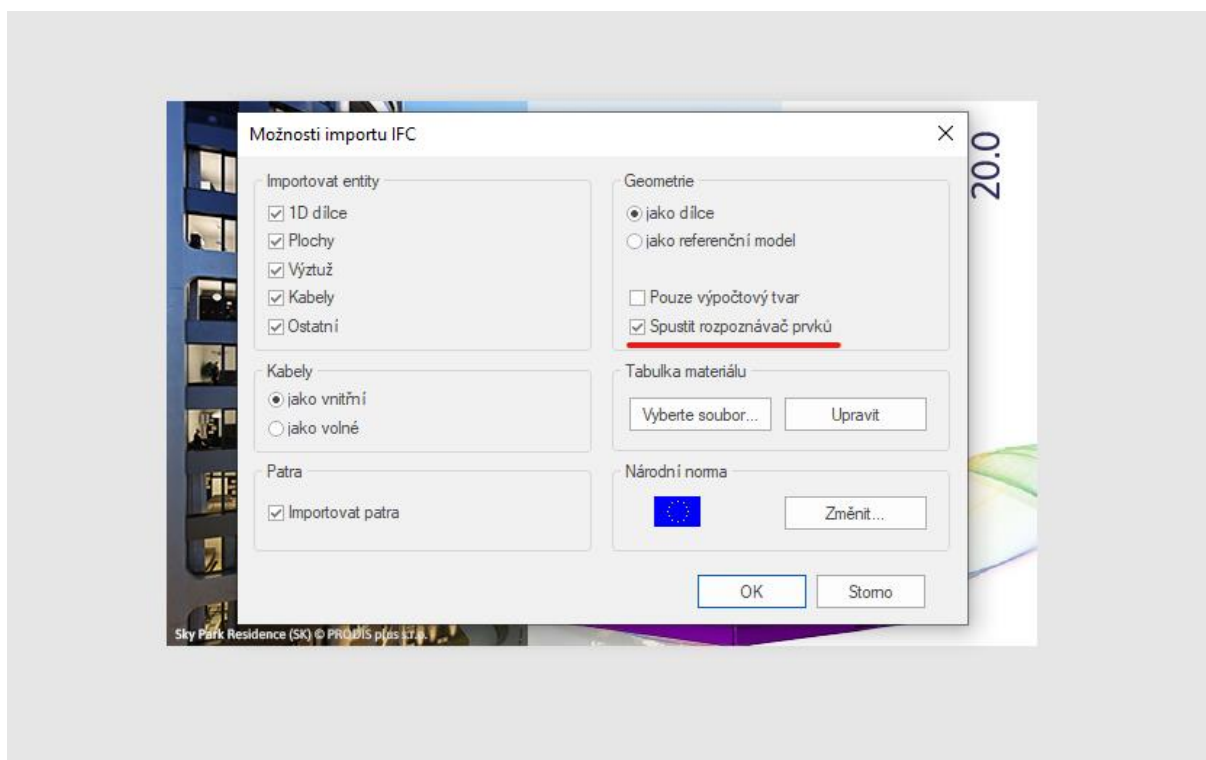
Obrázek 88 - Grafický x analytický model příčného rámu

Statických softwarů je na trhu spousta ale v rámci diplomové práce jsem si připravil import IFC modelu do softwaru Scia Engineer, s kterou mám nejvíce zkušenosti. Právě Scia má plugin přímo ušitý na míru pro převod Revit modelu do Scii. Nicméně tento plugin není aktuální pro poslední verzi Revitu, v kterém prvky modelují, a proto není možné ho využít.

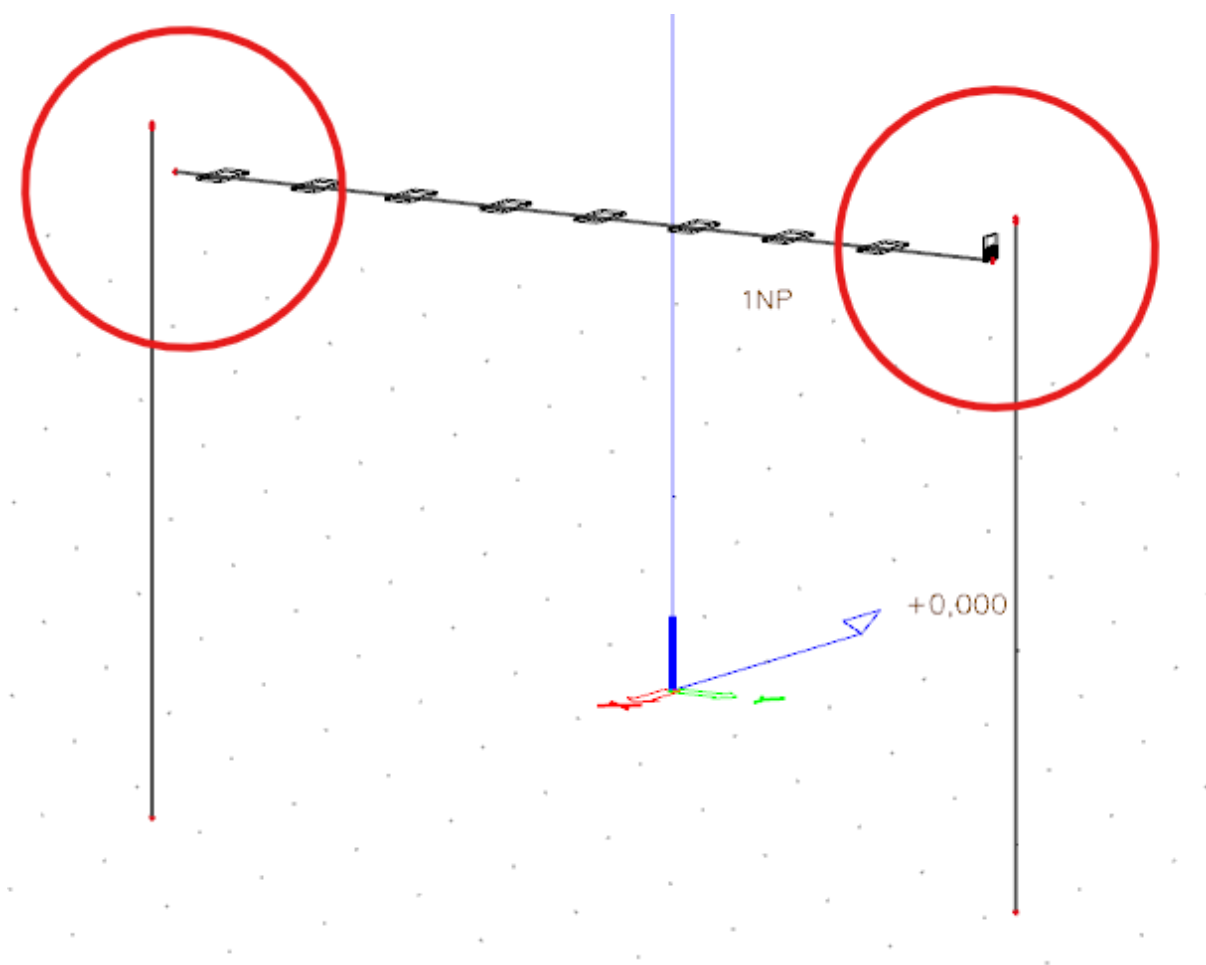
Budu tedy pokračovat přes import skrze IFC soubor. Tento proces není plně automatický a je možné upravit jeho nastavení viz obrázek 89. Při importu IFC souboru do Scii je možné a v našem případě i nutné zaškrtnutí možnosti spustit rozpoznávač prvků. Pokud tuto možnost nezaškrtneme, prvky se naimportují jako obecné modely a nebude na nich možné spustit statickou analýzu. Avšak pokud tuto možnost aktivujeme, model se v procesu upraví způsobem, že přeruší analytické vazby a není použitelný viz obrázek 90.

V mém případě se jedná pouze o jednu příčnou vazbu, ale kdybych exportoval celý objekt, množství úprav by bylo časově neekonomické, a tedy převod formátem IFC není řešením.

Formát IFC ve výsledku nebude použit jako převodní médium pro analytický model, jelikož k tomu není určený. Není to však do budoucna vyloučeno a je tu možnost růstu právě pro tento formát. Zároveň je nutné zmínit, že právě nastavení výstupu do formátu IFC je důležité a je nutné tento proces nezanedbat. V rámci mé práce jsem se ale tímto nezabýval, protože je to téma poměrně obsáhlé.



Obrázek 89 - Import IFC rámu do softwaru Scia Engineer

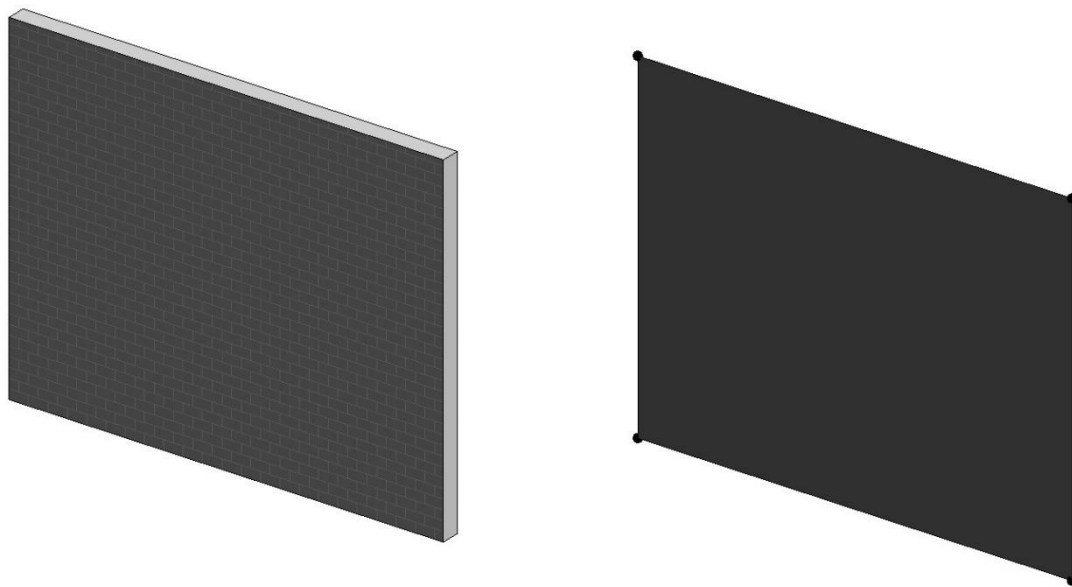


Obrázek 90 - Vazby rámu v softwaru Scia Engineer

### 5.3. Statická analýza v softwaru Autodesk Robot

Scia Engineer není zdaleka jediným softwarem na trhu. Pokud bychom se pohybovali na poli Autodesku, nabízí se software s názvem Robot Structural Analysis Professional krátce pouze Robot. Jelikož se jedná o stejného výrobce jako u Revitu, převod modelu mezi softwary funguje daleko lépe ale stále v našem případě poměrně nedokonale.

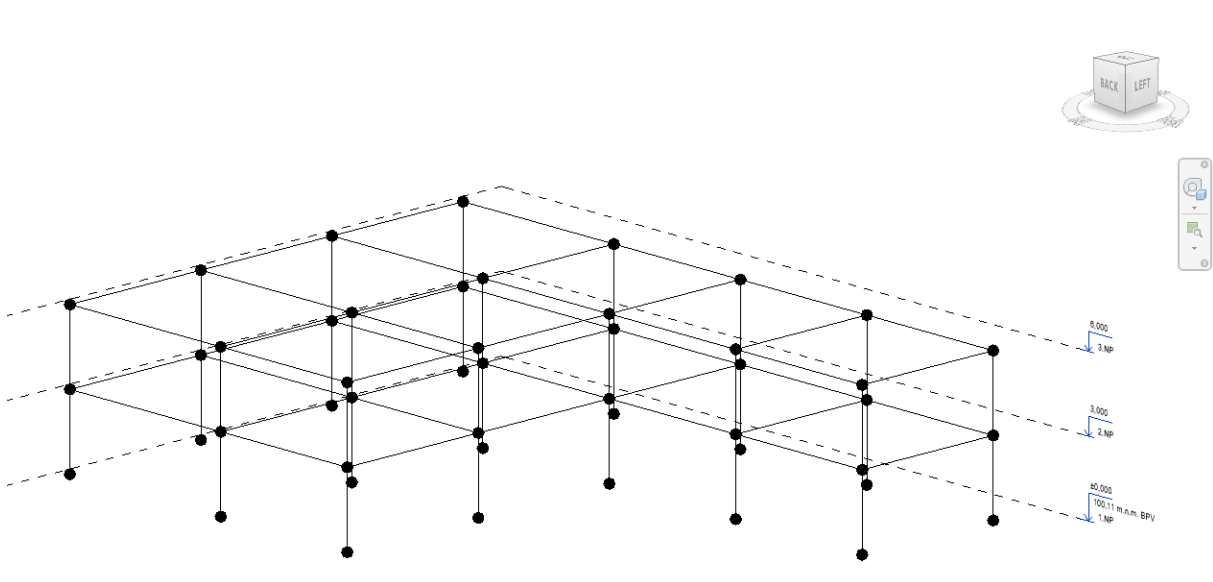
Převod mezi softwary funguje napřímo nebo do souboru smxx. Převod z Revitu do Robota může být snadný ale i tak se nevyhnu přípravám a úpravám modelu, pravděpodobně díky složitosti systému DPS. Co tím myslím se dá jednoduše znázornit na výztužné stěně, která nemá v modelu analytické zobrazení. Je v modelu pouze jako instance, která nese určité informace, ale jako prvek s analytickým zobrazením to není. U těchto prvků je tedy nutná úprava v Robotovi, a jejich následné doplnění. Jedná se sice o dvojitou práci, ale bohužel nám zatím software nenabízí mně známou jinou cestu. Určitě by se dala nasimulovat výztužná stěna s analytickým modelem, kdyby byla jako stěna modelována, avšak její analytický model by zase neodpovídal realitě. Můžete si totiž všimnout, že výztužná stěna má kotevní body ve vrcholu a patě na 2, 3 nebo více místech a kraje stěny se tak vůbec nedotýkají přilehlých prvků jako sloupy a nosníky. Jak vypadá analytický model stěny můžeme vidět na obrázku 91.



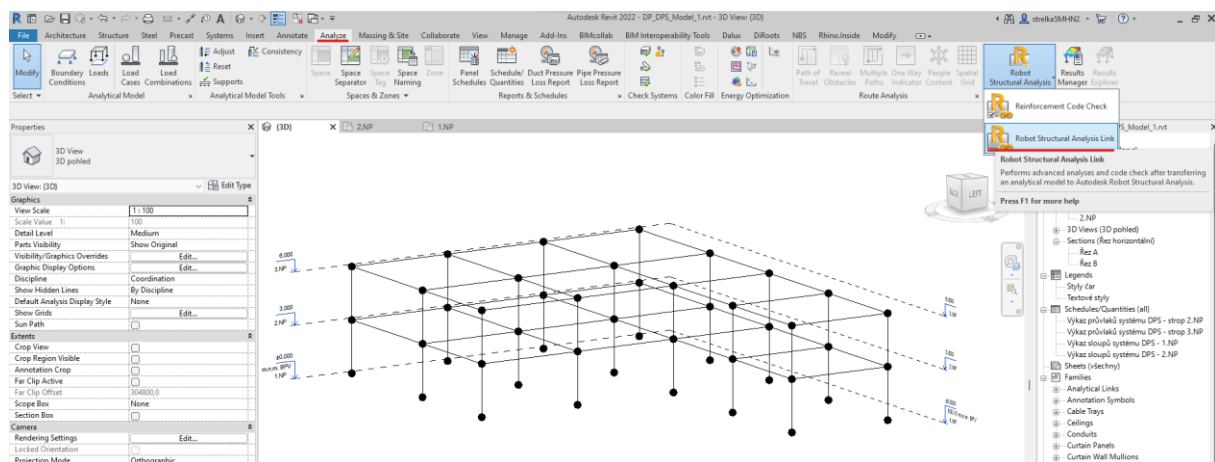
Obrázek 91 - Stěna grafický x analytický model

Převod modelu pomocí souboru .smxx neslouží přímo jako nativní formát softwaru Robot, avšak používá se, pokud dojde k úpravě modelu například v Robotovi a je třeba úpravy přenést do modelu v Revitu. Soubor .smxx je pak právě k tomuto využívám. V mém případě jsem toto nepoužil, jelikož bylo mým cílem primárně ukázat analytické vazby a propojení Revitu s Robotem. V mém případě jsem tedy postupoval cestou přímé přenosu modelu do Robota.

Analytický model mého objektu č. 1 můžeme vidět na obrázku 93. Tento model je definovaný úsečkami neboli osami prvků a koncovými body. Co tu však zázorně není, ale při převodu se taktéž převede jsou základy. Pokud pod sloupy umístím základové patky, jako jsme viděli na vzorovém modelu č. 1, Robot si patky přebere jako podpory a nemusím je tak zadávat ručně, maximálně budu muset upravit jejich stupně volnosti.



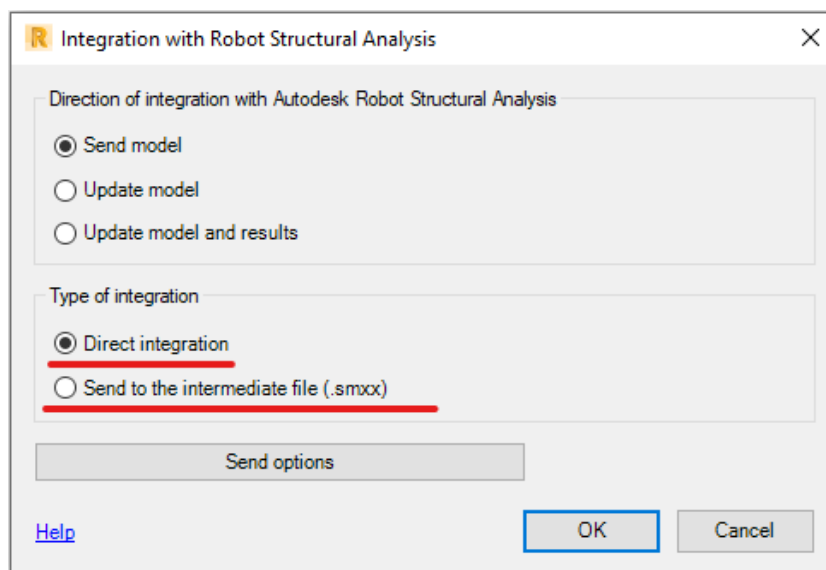
Obrázek 92 - Analytický model vzorového modelu č.1



Obrázek 93 - Export modelu do Robot Structural Analysis

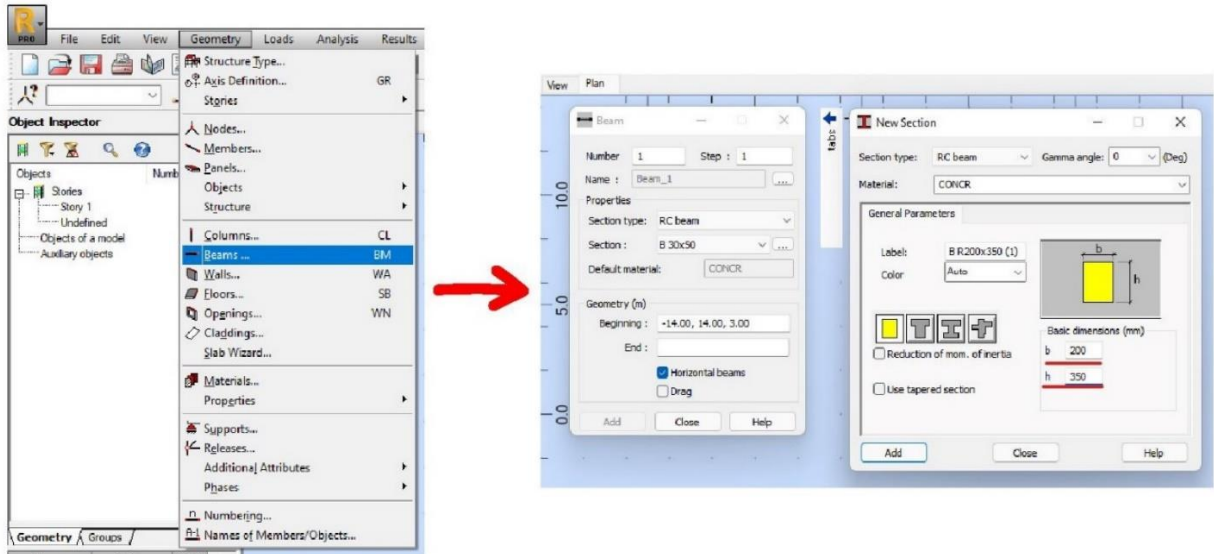
Pokud budeme postupovat systematicky, přes kartu analyzovat, nalezneme logo Robota, při rozbalení nám Robot nabídne dvě možnosti. V mém případě je správná právě ta druhá, která je na obrázku 92 podržena červeně. Důležité je mít taktéž otevřený současně software Robot, jelikož export neprobíhá do souboru ale přímo ze softwaru do softwaru.

Stejně jako Scia na při převodu Robot nabízí upravit nastavení přenosu modelu. V tomto případě je to však jednodušší a uživatel si tak pouze vybírá, za jakým účelem model exportuje nebo aktualizuje a zda se jedná o přímou integraci nebo přenos do souboru .smxx. Toto nastavení můžeme vidět na obrázku 94.

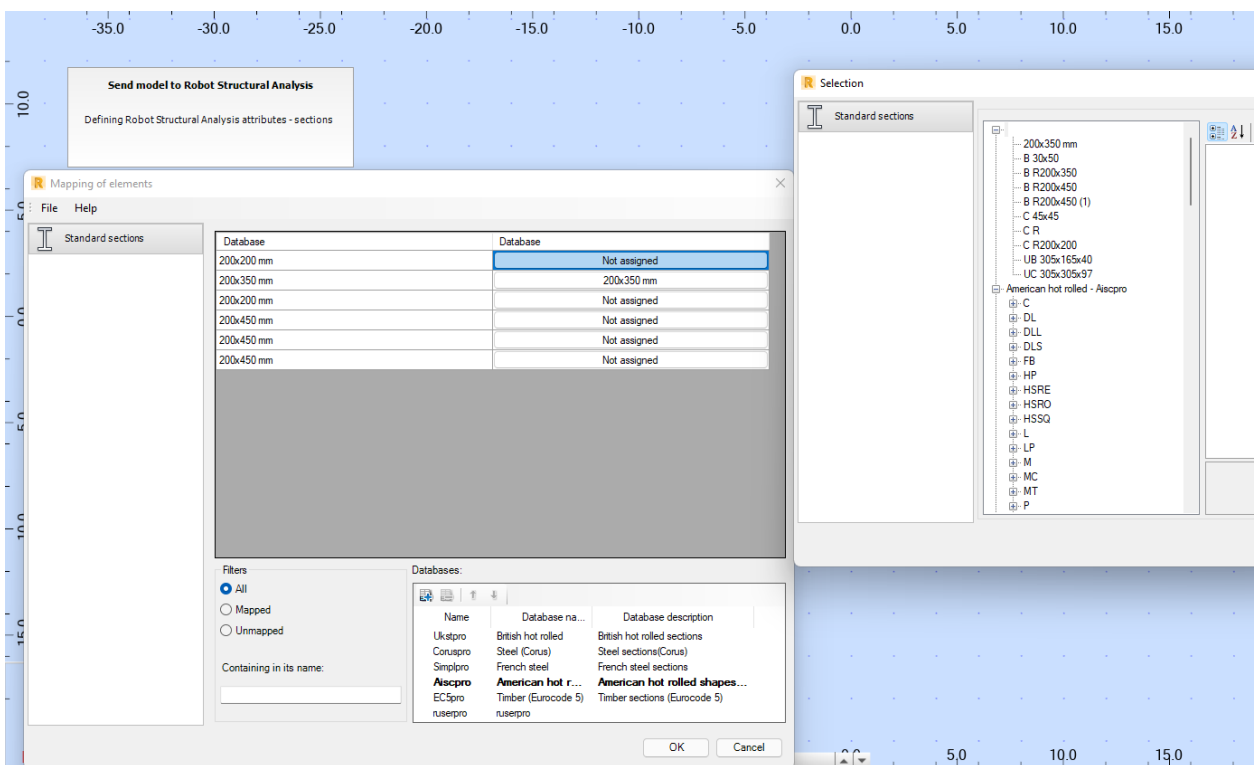


Obrázek 94 - Nastavení exportu do Robot

Co se však při přenosu nepřevéde, tedy alespoň u mých prvků je jejich profil a přesné grafické zobrazení včetně výřezu na kotvy a podobně. Pravděpodobně vzhledem k velkému množství zářezů a vložených rodin musíme při převodu prvkům přiřadit profil. Tyto profily je nutné si předem v Robotu nastavit a při převodu je přiřadit převáděným prvkům. Nastavení profilů můžeme vidět na obrázku 95 a přiřazení poté na obrázku 96.

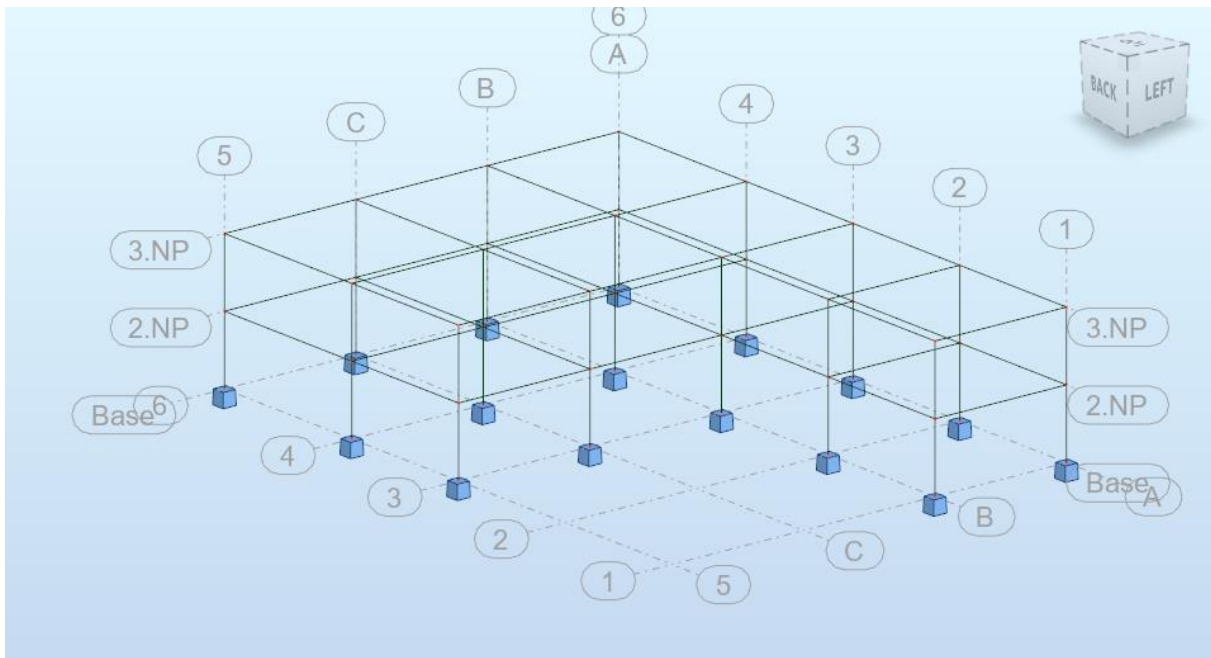


Obrázek 95 - Příprava profilů v Robotovi pro import modelu



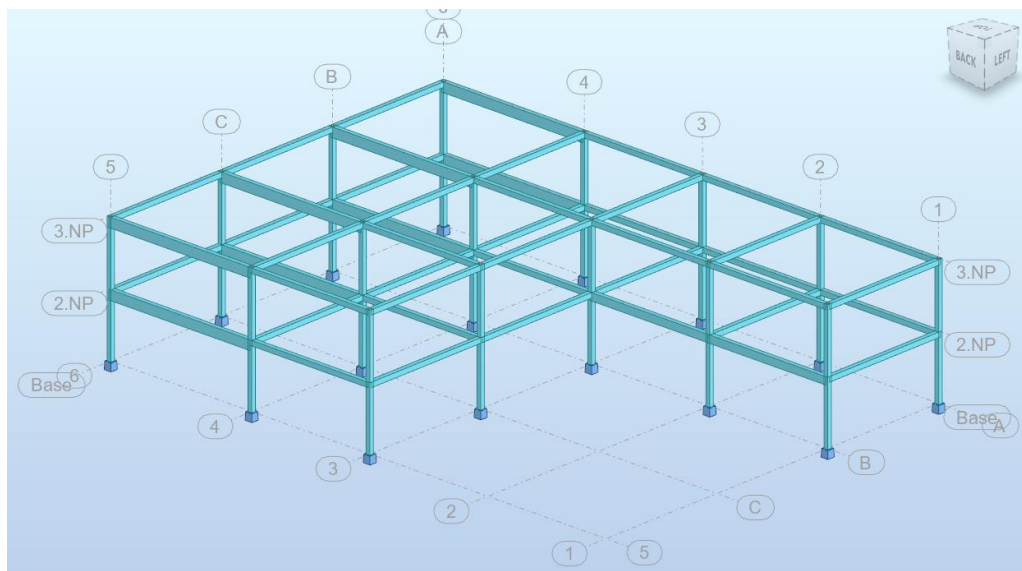
Obrázek 96 - Přřazení profilu referenčnímu prvku DPS

Po zatřídění profilů se nám model převede v analytické formě stejné jako v Revitu viz obrázek 97. V modelu je nutné ještě upravit zarovnání prvků vzhledem k jejich k ose, která po převodu nedopovídá modelu v Revitu.



Obrázek 97 - Analytický model v Robotu

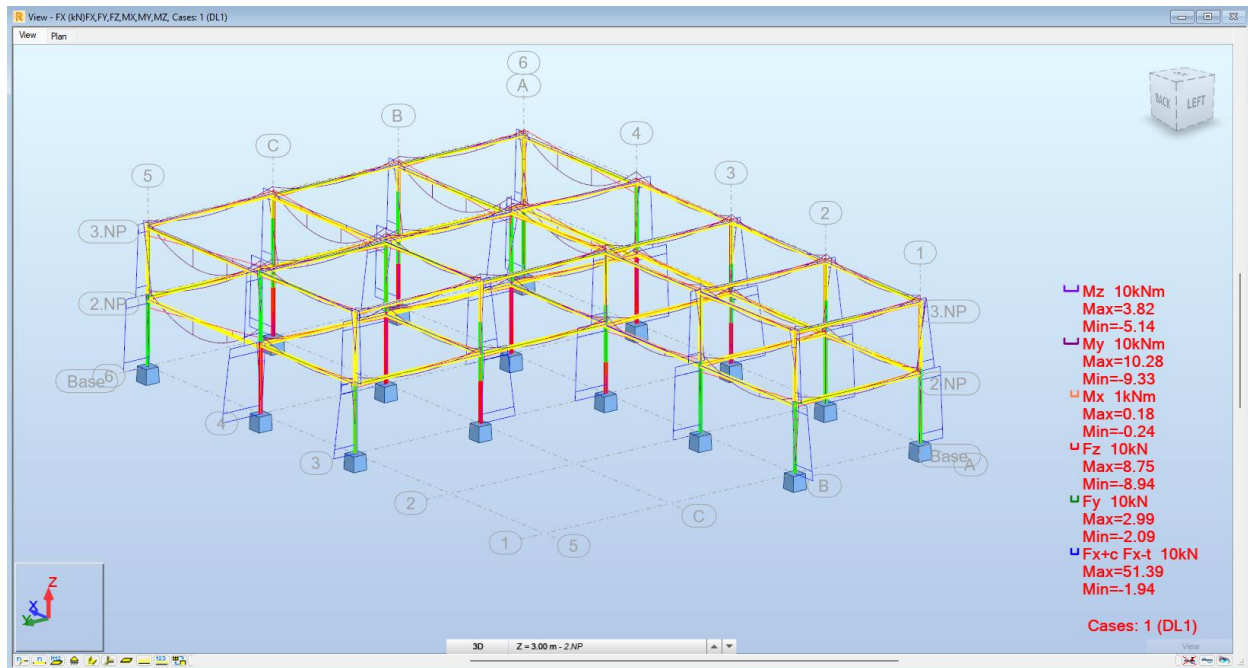
Jak můžete vidět na obrázku 98 nepřevodili se mi ani ztužující stěny, patníky nebo stropní nosníky spirall. Všechny tyto prvky je nutné do modelu doplnit, buď jako konstrukční prvky nebo zatížení.



Obrázek 98 - Grafický model v Robotu

Po načtení modelu a přiřazení profilů jednotlivým prvkům jsem pro kontrolu spustil na modelu statickou analýzu se zatížením pouze vlastní tíhou. Výsledky vnitřních svislých sil,

můžeme vidět na obrázku 99. Dle výsledků jsou nejvíce zatížené sloupky v místě souběhu 4 sloupů, což je správně. Největší tlakové síly jsou v patě sloupů a jejich nárůst je po délce sloupky lineární. Momenty jsou taktéž největší na nejdelších vodorovných prvcích. Můžeme si všimnout, že momenty se však nechovají jako na prostém nosníku a je třeba upravit i vazbu jednotlivých prvků jako třeba nosník sloup v jejich styčnicku.

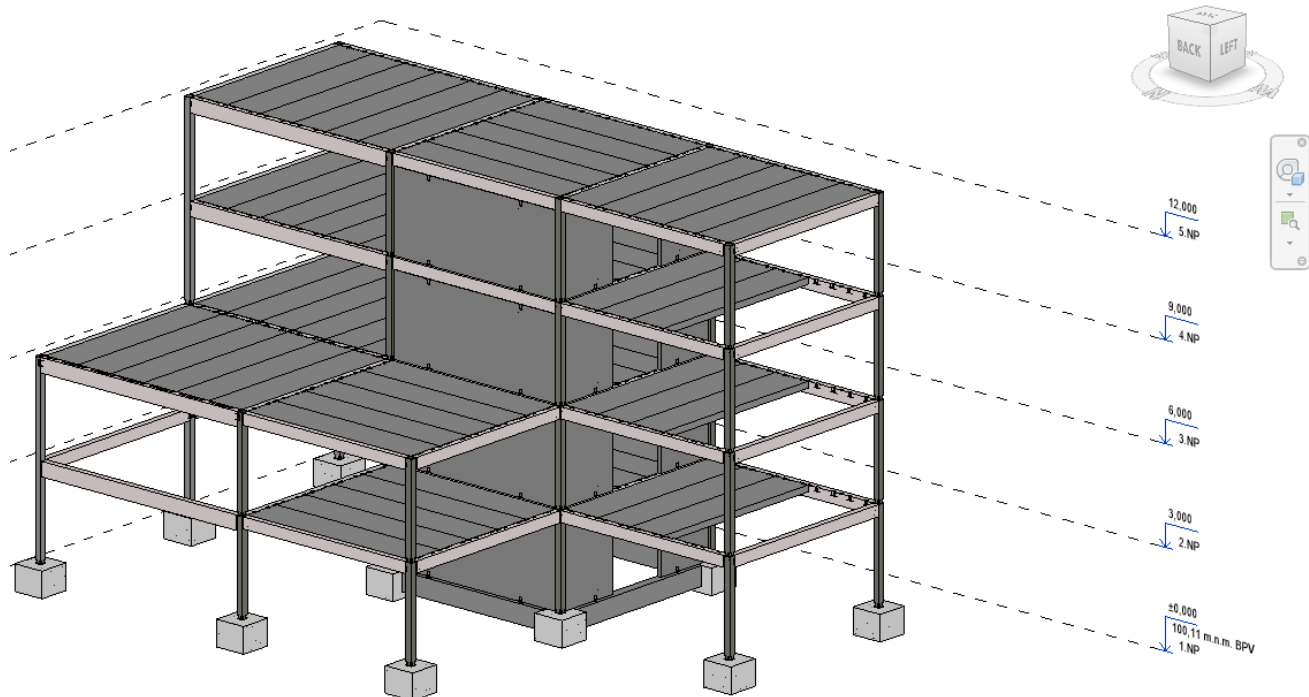


Obrázek 99 - Výpočet vnitřních sil od vlastní tíhy

Můžeme si všimnout, že jsou na některých prvcích vnitřní síly vykresleny jiným směrem, hlavně u tlakových sil, to je způsobeno natočením prvku v modelu, které následně určuje i lokální souřadnicový systém prvkům v Robotu. Lokální souřadnicový systém následně ovlivňuje vykreslení sil na prvcích.

Druhý model, tedy objekt č. 2 byl z části vytvořen ze stavebních prvků modelu č. 1 a doplněn o prvky, které byly pro budovu nezbytné. Model č. 2 můžeme vidět na obrázku 100.





Obrázek 100 - 3D model vzorového modelu č.2

#### 5.4. Použití klasifikačního systému a tvorba výkazu výměr a výrobků

Klasifikační systém v BIMu slouží jako značení stavebních prvků, které popisujeme řadou písmen a čísel. Klasifikační systémy jsou víceúrovňové a různé. V ČR máme klasifikační systém SNIM, který je součástí koncepce BIM od angetury ČAS pro ČR. Tento klasifikační systém používám i v mém projektu. SNIM neboli standart negrafických informací modelu, byl vyvinut organizací czBIM a má svoje pravidla používání. Tento klasifikační systém je sice stále ve stádiu vývoje ale v praxi je už použitelný a poměrně přehledný. Klasifikačních systémů na světě existuje několik, ale mezi nejznámější se řadí systémy Uniclass, Ominclass a Uniformat. Ominclass je klasifikační systém, který je v Revitu už implementován jako vestavěný parametr. Pokud chceme využít jiný klasifikační systém, řešíme to přes vestavěný parametr nazvaný jako kód sestavy. Abychom třídící systém v Revitu nastavily, musíme mít seznam v textovém formátu, který následně vkládáme přímo do Revitu. Jsou samozřejmě i jiné cesty, jako oficiální pluginy od Autodesku, které vám dle individuálního výběru klasifikační systém zatřídí k vybrané kategorii rodin. V našem případě jsme ale vybrali SNIM v menší uživatelské úpravě. Náš třídící systém je dvouúrovňový, například sloup je SL a prefabrikovaný 01, monolitický zase 02 a takto fungují i prvky ostatní.

Značení prvků v dokumentaci poté kombinuje tři parametry značení, tedy kód sestavy, typové označení a instanční označení. Pro náš projekt byla vytvořena speciální

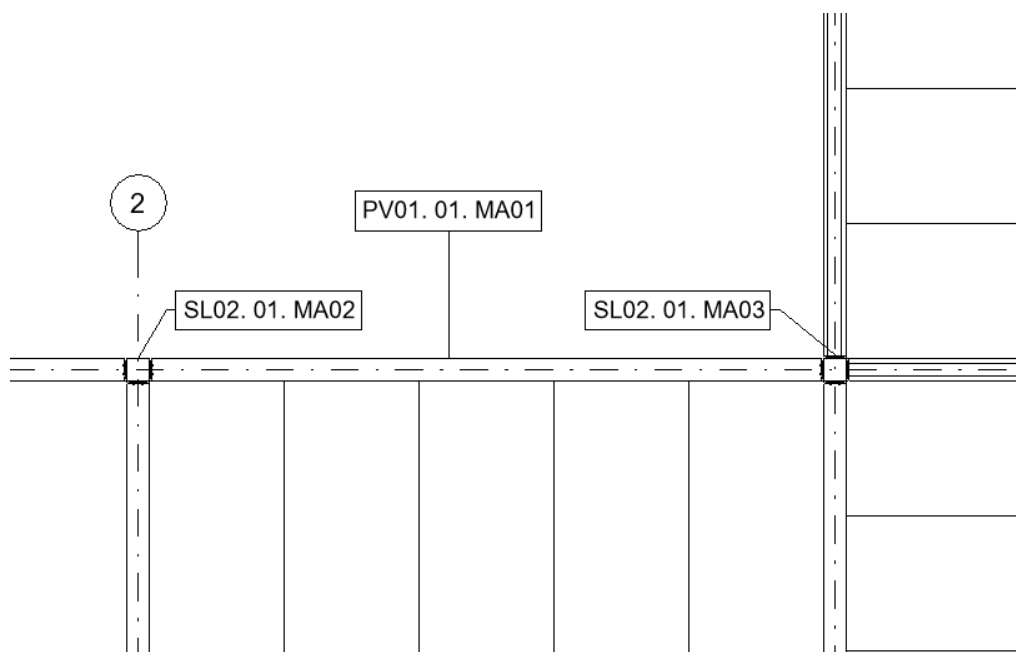
rodina popisku multikategorie, která kombinuje právě tyto tři parametry. Schéma označení můžeme vidět na obrázku 101.

**TM01.01.0001**

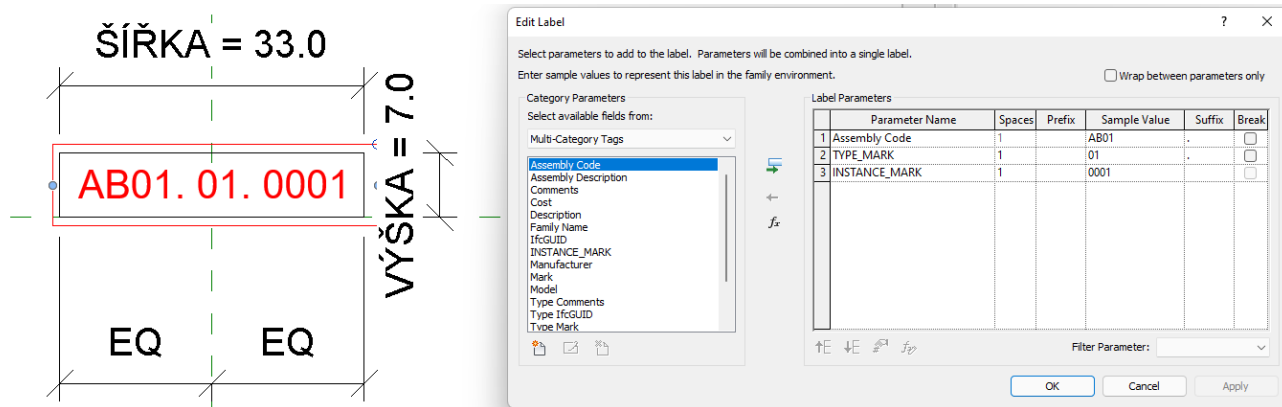
**KÓD**      **TYPOVÉ**    **INSTANČNÍ**  
**SESTAVY**    **OZNAČENÍ**    **OZNAČENÍ**

Obrázek 101 - Značení prvků DPS ve výkresové dokumentaci

Popisek multikategorie je poté rodina poznámky, která je vytvořena tak, aby kombinovala parametry výše znázorněné. Rodina funguje viz obrázek 102 a 103. Přímo v rodině popisku, pokud vytváříme text, který má referenci na určitý parametr, používáme prvek Label viz obrázek 103. Labelu se dá přímo přiřadit parametr, pokud je sdílený nebo vestavěný.



Obrázek 102 - Příklad značení prvků DPS



Obrázek 103 - Detail rodiny multikategorie

#### 5.4.1. Porovnání modelu č.1 a č.2

Model z prvků DPS nemusí sloužit pouze v procesu tvorby dokumentace pro DPS nebo DSP ale i po čas životnosti stavby a popřípadě demontáže a zase následné montáže. Proto jsem vytvořil druhý model, jiného účelu, rozmístění ale ze stejných prvků, jaké byly použity v modelu č. 1. Původní prvky však plně nestačily na vybudování modelu č.2 a byly tam tak přidány i prvky nové. Zároveň si na prvcích představíme, jak funguje třídící systém v závislosti na jejich původním modelovém zdroji, jak jsme již popisoval výše.

Dále jsem vytvořil výkaz celkového množství prvků v modelu č. 2, které jsem rozdělil na část MA a MB. Tedy prvky z původního modelu a prvky z modelu nového viz obrázek 104 a 105. Data z výkazu jsem vložil do excelu a vytvořili graf, který ukazuje, kolik prvků bylo použito nových v poměru k prvkům, které se použili znovu a byly recyklovány tím nejlepším možným způsobem, tedy znovu využity.

# PRVKY Z MODELU Č. 1

Výkaz průvlaků systému DPS - CELKEM MA						
s	Označení	Šířka profilu	Výška profilu	Délka nosníku	Počet	Objem
BEAM_CLEAN	PV01.01.MA01	350	200	5960	6	2,50 m <sup>3</sup>
BEAM_CLEAN	PV01.02.MA01	450	200	7160	1	0,64 m <sup>3</sup>
BEAM_CLEAN	PV01.01.MA01	350	200	5960	6	2,50 m <sup>3</sup>
BEAM_CLEAN	PV01.01.MA01	350	200	5960	3	1,25 m <sup>3</sup>
BEAM_CLEAN	PV01.01.MA01	350	200	5960	3	1,25 m <sup>3</sup>
BEAM_INNER	PV01.01.MA01	450	200	7160	1	0,62 m <sup>3</sup>
BEAM_OUTER	PV01.01.MA01	350	200	5960	6	2,45 m <sup>3</sup>
BEAM_OUTER	PV01.01.MA01	350	200	5960	6	2,45 m <sup>3</sup>
BEAM_OUTER	PV01.02.MA01	450	200	7160	1	0,63 m <sup>3</sup>
BEAM_OUTER	PV01.01.MA01	350	200	5960	4	1,64 m <sup>3</sup>
BEAM_OUTER	PV01.01.MA01	350	200	5960	4	1,64 m <sup>3</sup>
BEAM_OUTER	PV01.02.MA01	450	200	7160	1	0,63 m <sup>3</sup>
					42	18,21 m <sup>3</sup>

Výkaz sloupů systému DPS - CELKEM MA						
Rodina	Označení	Šířka profilu	Výška profilu	Délka sloupu	Počet	Objem
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA02	200	200	3000	2	0,23 m <sup>3</sup>
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA03	200	200	3000	2	0,23 m <sup>3</sup>
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA02	200	200	3000	2	0,23 m <sup>3</sup>
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA03	200	200	3000	2	0,23 m <sup>3</sup>
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA01	200	200	3000	3	0,35 m <sup>3</sup>
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA02	200	200	3000	3	0,35 m <sup>3</sup>
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA01	200	200	3000	3	0,35 m <sup>3</sup>
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MA02	200	200	3000	3	0,35 m <sup>3</sup>
					20	2,34 m <sup>3</sup>
COLUMN_RECTANGULAR_TWOLEVEL	SL02.01.MA01	200	200	6000	1	0,24 m <sup>3</sup>
COLUMN_RECTANGULAR_TWOLEVEL	SL02.01.MA02	200	200	6000	1	0,24 m <sup>3</sup>
COLUMN_RECTANGULAR_TWOLEVEL	SL02.01.MA01	200	200	6000	1	0,24 m <sup>3</sup>
					3	0,71 m <sup>3</sup>

Výkaz stropních nosníků spiroll - CELKEM MA				
Rodina	Označení	Výška profilu	Počet	Objem
SPIROLL	SK02.01.MA01	200	68	51,73 m <sup>3</sup>
			68	51,73 m <sup>3</sup>

Výkaz ztužujících stěn DPS - CELKEM MA				
Rodina	Označení	Výška profilu	Šířka profilu	Počet
DIAPHRAGM	SN01.01.MA01	2650	120	4
DIAPHRAGM	SN01.01.MA02	2650	120	4

Výkaz základových pasů DPS - CELKEM MA						
Rodina	Označení	Šířka profilu	Výška profilu	Délka pasu	Počet	Objem
CURB	ZS03.01.MA01	300	600	5200	4	3,74 m <sup>3</sup>
					4	3,74 m <sup>3</sup>

Obrázek 104 - Výkaz prvků z modelu č.1

## PRVKY Z MODELU Č. 2

Výkaz průvlaků systému DPS - CELKEM MB						
Rodina	Označení	Šířka profilu	Výška profilu	Délka nosníku	Počet	Objem
BEAM_CLEAN	PV01.02.MB01	450	200	7160	1	0,64 m <sup>3</sup>
BEAM_OUTER	PV01.02.MB01	450	200	7360	1	0,57 m <sup>3</sup>
BEAM_OUTER	PV01.01.MB01	350	200	5960	1	0,41 m <sup>3</sup>
BEAM_OUTER	PV01.02.MB01	450	200	7160	2	1,27 m <sup>3</sup>
BEAM_OUTER	PV01.01.MB01	350	200	5960	1	0,41 m <sup>3</sup>
BEAM_OUTER	PV01.02.MB01	450	200	7160	1	0,63 m <sup>3</sup>
					7	3,93 m <sup>3</sup>

Výkaz sloupů systému DPS - CELKEM MB							
Rodina	Označení	Šířka profilu	Výška profilu	Délka sloupu	Počet	Objem	
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MB01	200	200	3000	1	0,12 m <sup>3</sup>	
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MB02	200	200	3000	3	0,35 m <sup>3</sup>	
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MB01	200	200	3000	2	0,23 m <sup>3</sup>	
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MB02	200	200	3000	2	0,24 m <sup>3</sup>	
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MB02	200	200	3000	1	0,12 m <sup>3</sup>	
COLUMN_RECTANGULAR_SINGLELEVEL	SL02.01.MB02	200	200	3000	1	0,12 m <sup>3</sup>	
					10	1,18 m <sup>3</sup>	
COLUMN_RECTANGULAR_TWOLEVEL	SL02.01.MB01	200	200	6000	1	0,24 m <sup>3</sup>	
					1	0,24 m <sup>3</sup>	

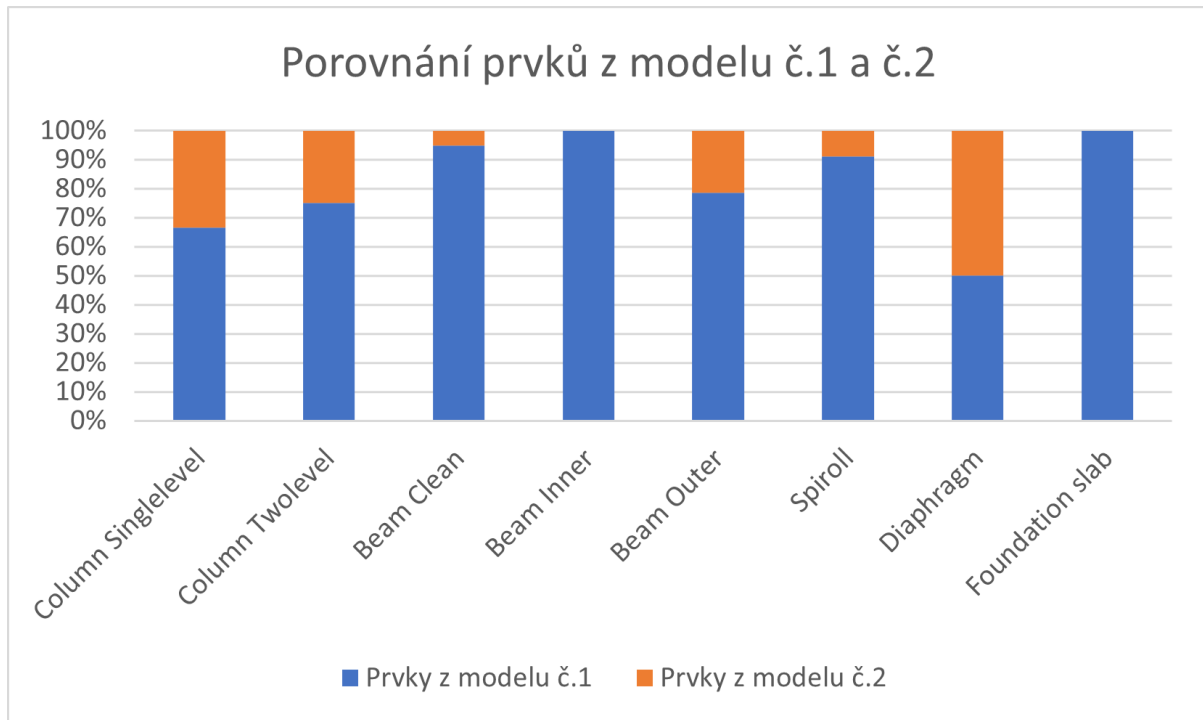
Výkaz stropních nosníků spiroll - CELKEM MB				
Rodina	Označení	Výška profilu	Počet	Objem
SPIROLL	SK02.01.MB01	200	6	4,56 m <sup>3</sup>
			6	4,56 m <sup>3</sup>

Výkaz ztužujících stěn DPS - CELKEM MB				
Rodina	Označení	Výška profilu	Šířka profilu	Počet
DIAPHRAGM	SN01.01.MB01	2650	120	4
DIAPHRAGM	SN01.01.MB02	2650	120	4

Obrázek 105 - Výkaz prvků z modelu č.2

Na obrázku 106 poté můžeme vidět graf, který představuje poměr mezi použitými prvky z modelu 1 a 2. Graf je založen na počtu prvků, které jsou použity. Největší množství prvků, o které musel být model č. 2 doplněn jsou ztužující stěny, které vzhledem ke zvýšení pater objektu muselo být nově vytvořeno 50% z celkového množství.

Na vodorovné ose grafu vidíme jednotlivé prvky, které byly v modelu použity, na svislé ose poté stupnici v procentech do 100%. Modře jsou označeny prvky z modelu č. 1 a oranžově poté prvky z modelu č. 2.



Obrázek 106 - Porovnání prvků z modelu 1 a modelu 2

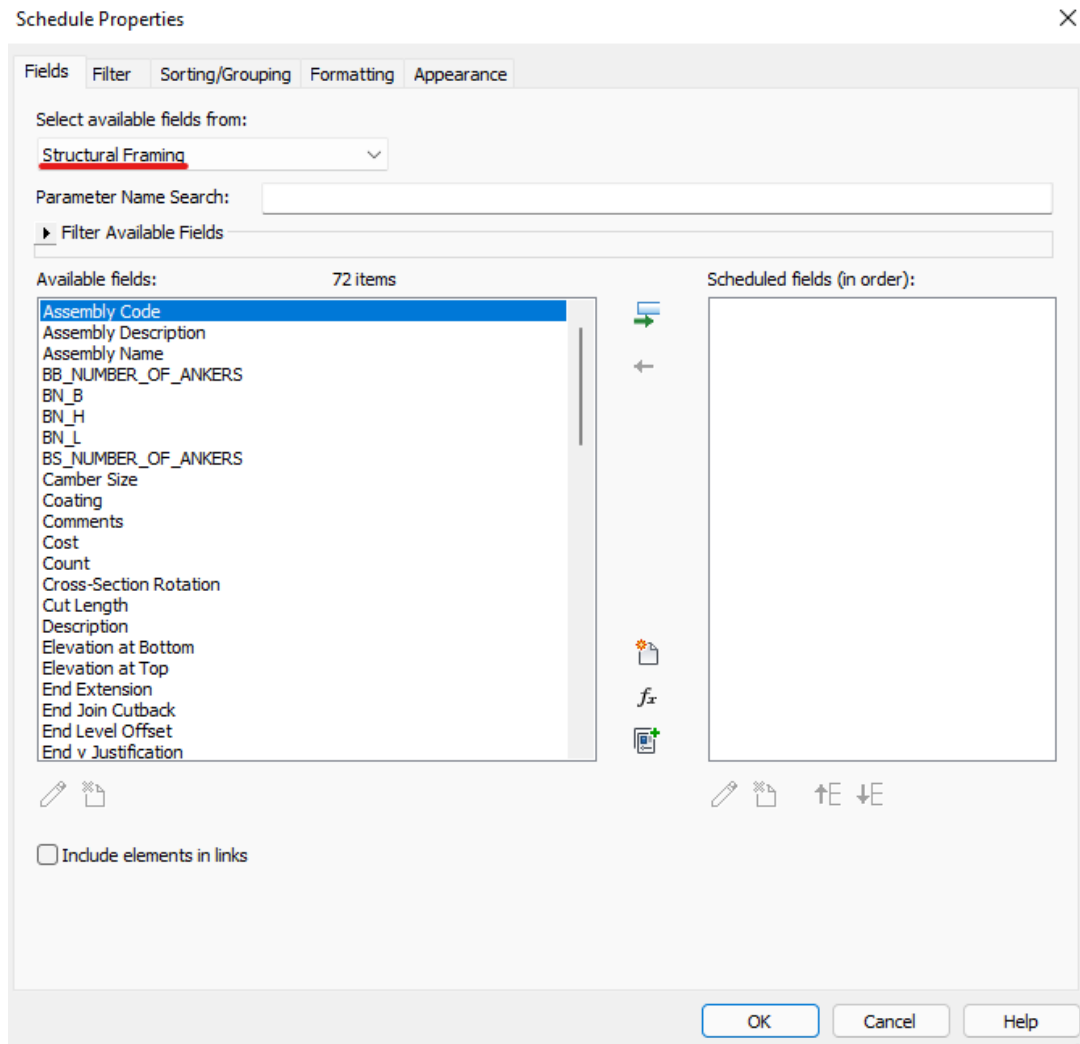
Z grafu je viditelné, že na model č. 2, který má rozdílné využití, bylo použito v průměru zhruba 80% prvků z modelu č. 1. To znamená, že i rozpočet na výrobu prvků, by byl značně redukován, avšak tam by bylo nutné ještě započítat práce spojené s demontáží, přesunem a montáží.

## 5.5. Tvorba statických výkresů

Tvorba dokumentace v Revitu je trochu něco jiného, než tvorba dokumentace ve 2D kreslících programech jako je například Autocad. Vše je v Revitu modelováno a pohledy ať už 3D nebo půdorysné či řezy generují vždy aktuální grafickou realitu modelu. Pro diplomovou práci byly vygenerovány 2 půdorysné pohledy na strop z modelu č.1 a jednotlivé prvky byly označeny popiskou multikategorie. K těmto půdorysům byly následně vygenerovány výkazy, které přímo dle značení odkazují na určité prvky modelu, a tak i v dokumentaci.

Výkazy v Revitu taktéž fungují na několika principech. Pokud budeme vykazovat jednotlivé prvky, budeme vykazovat dle kategorie, tedy konstrukční rámová konstrukce nebo

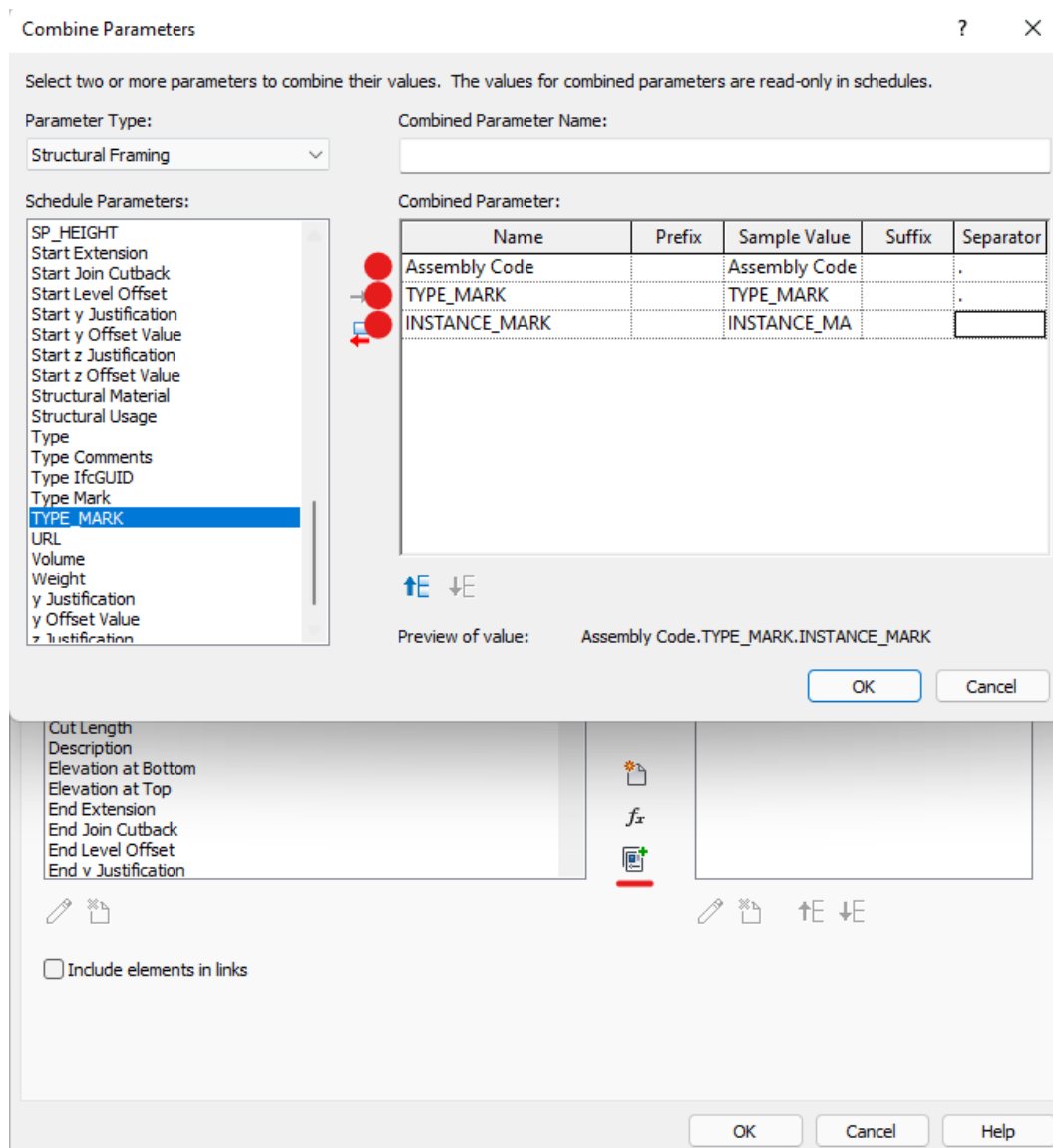
konstrukční sloupy. Můžeme ale například vykazovat pouze materiál. Po výběru kategorie vyskočí okénko s výběrem parametrů, které chceme vykazovat viz obrázek 107. Jelikož jsou všechny parametry sdílené, objeví se nám v tomto seznamu, pokud by sdílené nebyly, tyto parametry bychom tam nenašli.



Obrázek 107 - Tvorba výkazu prvků DPS

Parametry se taktéž dají kombinovat do jednoho parametru nebo se dají vložit do rovnice, pokud jsou to parametry číselné. Právě tuto první možnost spojení parametrů používám ve výkazu, aby označení prvků vypadalo tak jako na obrázku 101.

Výsledkem jsou tedy dva půdorysné pohledy na strop, se základními kótami, označením prvků s využitím výše zmíněné kombinace a výkazy výměr, které definují prvek dle označení a jeho další parametry. Tyto výkresy jsou součástí diplomové práce jako příloha.



Obrázek 108 - Kombinované parametry

Pokud budeme chtít vykázat kotvy, proces je podobný tomu s nosnými prvky, který je naznačen výše. Rozdíl je v tom, že kotvy jsou kategorie rodiny konstrukční přípoje a podle této kategorie musí být i vykázány.

Parametry kotev jsou sdílené a je tedy možné je vykázat. Nejedná se pouze o parametry určující rozměry kotvy ale i parametry určující materiál kotvy nebo její referenční označení. Referenční označení určuje, k jakému prvku kotva přiléhá. Jedná se o stejné označení, jaké používám u nosných prvků, které se skládá z kódu sestavy, typového označení a označení instančního. Rozdíl je akorát v tom, že se označení vyplňuje automaticky na základě označení prvku.

Parametry kotev se tedy dají vykázat, ale jelikož se jedná o kotvy poměrně složité, musí být k čemu tyto parametry přiřadit. Původní myšlenka byla vytvořit dokumentaci kotev



z modelu s vykázanými parametry. Nicméně v procesu tvorby práce vznikl i grafický manuál nebo spíše katalog prvků DPS, který je součástí přílohy. Dokument slouží ke spojení parametru a grafického zobrazení kotvy. V tomto případě pak není nutné, vytvářet dokumentaci kotev přímo ale stačí pouze výkaz parametrů, které skrze katalog prvků, zatřídí té správné referenční vzdálenosti.

Pokud by však byl požadavek na dokumentaci kotev, jsem schopen v rámci modelu kotvy vyfiltrovat a vytvořit dokumentaci s výkresy pro jednotlivé kotvy. Výkaz parametrů kotvy z modelu č. 1 můžeme vidět na obrázku 109, kde jsou vykázané parametry kotvy sloup nosník.

Rodina	Referenční označení	Počet	CN_CB_ANKER_EXCENTRICITY	CN_CB_ANKER_LENGTH	CN_CB_ANKER_NOTCH
CB Connection	SL02.01.MA01	20	40	20	30
CB Connection	SL02.01.MA02	46	40	20	30
CB Connection	SL02.01.MA03	24	40	20	30

Structural Connection Column Beam

CN_CB_ANKER_THICKNESS	CN_CB_ANKER_WIDTH	CN_CB_ANKER_WIDTH2	CN_CB_HEIGHT	CN_CB_HEIGHT2	CN_CB_REBAR_DIAMETER
10	10	20	150	120	6
10	10	20	150	120	6
10	10	20	150	120	6

CN_CB_REBAR_EXCENTRICITY	CN_CB_REBAR_LENGTH	CN_CB_THICKNESS	CN_CB_THICKNESS2	CN_CB_WIDTH	CN_CB_WIDTH2
40	50	20	10	150	120
40	50	20	10	150	120
40	50	20	10	150	120

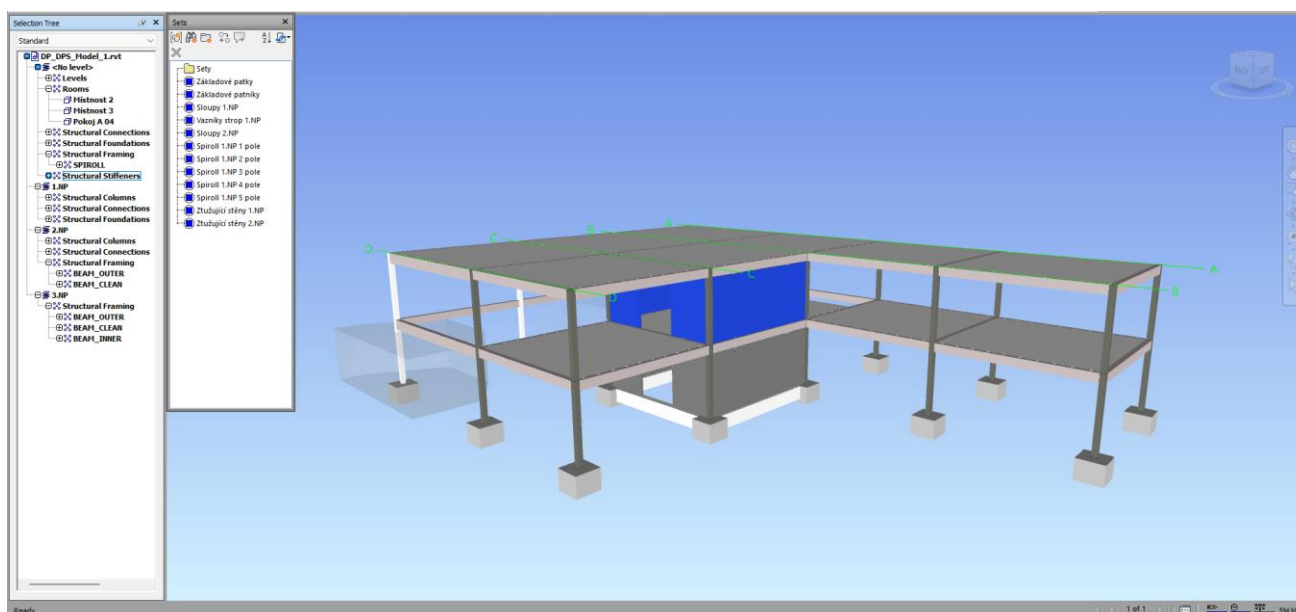
Obrázek 109 - Parametry kotvy sloup nosník

## 6. Další využití modelu z dynamických prvků DPS

Abych předvedl, k čemu je dále uživatel schopen model využívat, převedu svůj vzorový model do softwaru Autodesk Navisworks, vytvořím zjednodušený harmonogram a animaci výstavby v závislosti na harmonogramu. Autodesk nabízí software Navisworks ve třech variantách, tedy Manage, Freedom a Simulate. Tyto verze se liší cenou a jejich nástroji. V mojí ukázce budeme využívat Navisworks Manage.

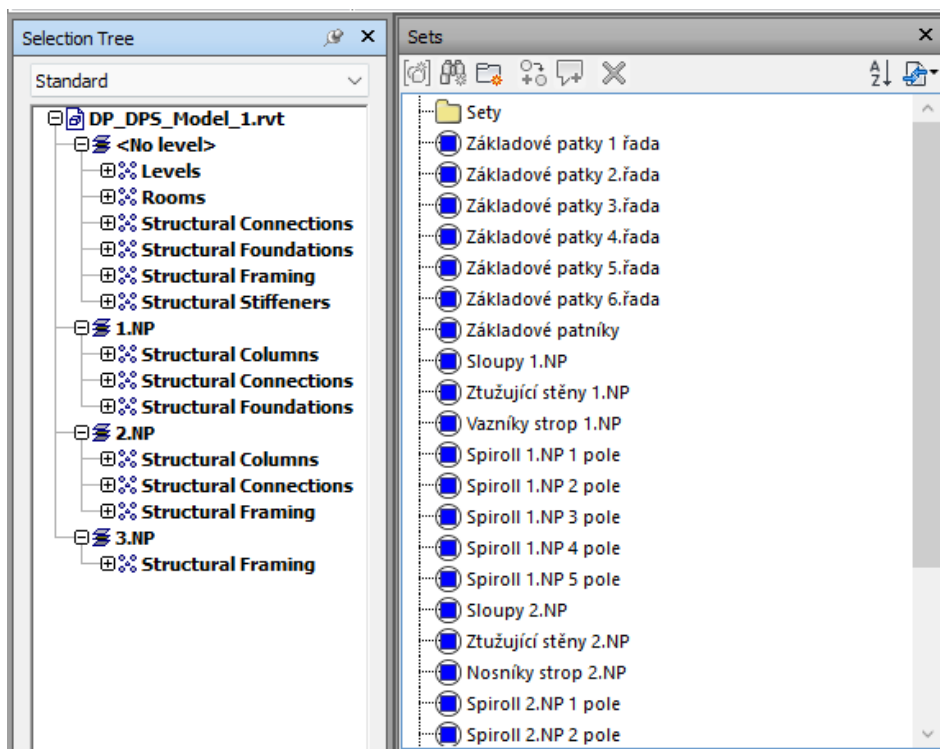
### 6.1. Příklad tvorby harmonogramu a animace výstavby

Převod modelu rvt. do Navisworks Manage neprobíhá stejně jako u Robota, nýbrž stačí otevřít Navisworks, příkaz otevřít soubor, změnit typ souboru na rvt., najít typový model a otevřít. Soubor následně uložíme jako Navisworks soubor s koncovkou .nwf nebo .nwc. Model se nám otevře ve stejné detailnosti jako v Revitu, se všemi prvky i jejich komponenty viz obrázek 110.



Obrázek 110 - 3D model vzorového modelu č.1 v softwaru Navisworks

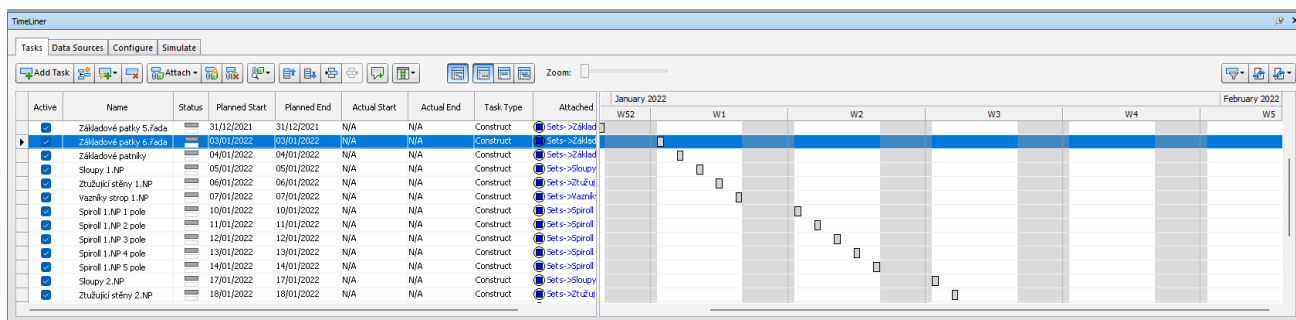
Rodiny se naimportují a dají se jednoduše procházet pomocí funkce selection tree. Selection tree funguje obdobně jako projektový prohlížeč v Revitu. Jsem schopen vybírat rodiny v celém projektu, nebo dle pater viz obrázek 111. Rodiny se taktéž dělí dle kategorie.



Obrázek 111 - Vytvořené sety modelu č.1

Na obrázku 111 taktéž vidíme funkci sets, která umožňuje prvky vkládat do skupin, tedy tak vytvářet skupiny prvků o 1 a více instancích. Tyto sety následně můžeme využít při tvorbě harmonogramu, který bude brát set jako jeden stavební proces a není tak třeba přiřazovat časový interval pouze jednomu prvku. V mém případě jsem rozdělil patky dle jednotlivých os, sloupy, nosníky, ztužující stěny dle patra a Spiroll panely dle polí.

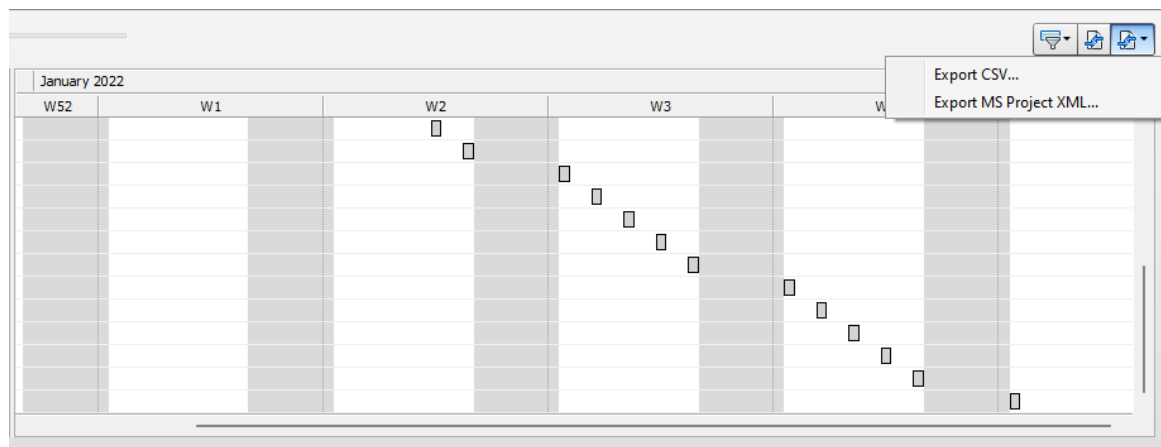
Zároveň je možné harmonogram dle setů vytvořit automaticky. Navisworks tak vezme každý set zvlášť a přiřadí mu jednu denní směnu, která dle nastavení trvá 8 hodin. Takhle sety rozdělí na dny mimo víkendy. Co se týče pracovní doby a práce přes víkendy, dá se tento parametr v Navisworks taktéž přednastavit. V mém případě jsem použil právě tento automatický proces bez úpravy parametrů délky směny a práce přes víkendy. Výsledný zjednodušený harmonogram můžeme vidět na obrázku 112, který obsahuje jednotlivé pracovní úkony, se dnem započatí práce, dnem ukončení, typem úkonu a přiřazeným setem.



Obrázek 112 - Automatická tvorba harmonogramu

Harmonogram taktéž slouží při výstavbě ke kontrole a dodržování harmonogramu, jelikož má kolonku na reálný začátek a konec pracovního procesu, kterou můžeme porovnávat se začátkem a koncem procesu naplánovaným. Na obrázku 112 jako Planned start a Actual start.

Harmonogram nabízí možnost exportu do souboru .csv nebo .xml, tedy formáty editovatelné v Microsoft Excel. Pro můj případ jsem předvedl export do souboru csv. a následně vložil do softwaru Excel viz obrázky 113 a 114. Tento export funguje i opačně, tedy jako import. Do Navisworks jsme schopni harmonogram naimportovat ve formátu .csv a na jeho základě tak upravit nebo vytvořit harmonogram nový.



Obrázek 113 - Export harmonogramu do .CSV souboru

Active	Name	Task Nesting	Actual Dates Enabled	Actual Start	Actual End	Planned Dates	Planned Start	Planned End	Task Type	Attached
1	1 Základové patky 1 řada	0	1			1	27/12/2021 09:00	27/12/2021 17:00	Construct	Sets->Základové patky 1 řada
2	1 Základové patky 2.řada	0	1			1	28/12/2021 09:00	28/12/2021 17:00	Construct	Sets->Základové patky 2.řada
3	1 Základové patky 3.řada	0	1			1	29/12/2021 09:00	29/12/2021 17:00	Construct	Sets->Základové patky 3.řada
4	1 Základové patky 4.řada	0	1			1	30/12/2021 09:00	30/12/2021 17:00	Construct	Sets->Základové patky 4.řada
5	1 Základové patky 5.řada	0	1			1	31/12/2021 09:00	31/12/2021 17:00	Construct	Sets->Základové patky 5.řada
6	1 Základové patky 6.řada	0	1			1	03/01/2022 09:00	03/01/2022 17:00	Construct	Sets->Základové patky 6.řada
7	1 Základové patníky	0	1			1	04/01/2022 09:00	04/01/2022 17:00	Construct	Sets->Základové patníky
8	1 Sloupy 1.NP	0	1			1	05/01/2022 09:00	05/01/2022 17:00	Construct	Sets->Sloupy 1.NP
9	1 Ztužující stěny 1.NP	0	1			1	06/01/2022 09:00	06/01/2022 17:00	Construct	Sets->Ztužující stěny 1.NP
10	1 Vazníky strop 1.NP	0	1			1	07/01/2022 09:00	07/01/2022 17:00	Construct	Sets->Vazníky strop 1.NP
11	1 Spiroll 1.NP 1 pole	0	1			1	10/01/2022 09:00	10/01/2022 17:00	Construct	Sets->Spiroll 1.NP 1 pole
12	1 Spiroll 1.NP 2 pole	0	1			1	11/01/2022 09:00	11/01/2022 17:00	Construct	Sets->Spiroll 1.NP 2 pole
13	1 Spiroll 1.NP 3 pole	0	1			1	12/01/2022 09:00	12/01/2022 17:00	Construct	Sets->Spiroll 1.NP 3 pole
14	1 Spiroll 1.NP 4 pole	0	1			1	13/01/2022 09:00	13/01/2022 17:00	Construct	Sets->Spiroll 1.NP 4 pole
15	1 Spiroll 1.NP 5 pole	0	1			1	14/01/2022 09:00	14/01/2022 17:00	Construct	Sets->Spiroll 1.NP 5 pole
16	1 Sloupy 2.NP	0	1			1	17/01/2022 09:00	17/01/2022 17:00	Construct	Sets->Sloupy 2.NP
17	1 Ztužující stěny 2.NP	0	1			1	18/01/2022 09:00	18/01/2022 17:00	Construct	Sets->Ztužující stěny 2.NP
18	1 Nosníky strop 2.NP	0	1			1	19/01/2022 09:00	19/01/2022 17:00	Construct	Sets->Nosníky strop 2.NP
19	1 Spiroll 2.NP 1 pole	0	1			1	20/01/2022 09:00	20/01/2022 17:00	Construct	Sets->Spiroll 2.NP 1 pole
20	1 Spiroll 2.NP 2 pole	0	1			1	21/01/2022 09:00	21/01/2022 17:00	Construct	Sets->Spiroll 2.NP 2 pole
21	1 Spiroll 2.NP 3 pole	0	1			1	24/01/2022 09:00	24/01/2022 17:00	Construct	Sets->Spiroll 2.NP 3 pole
22	1 Spiroll 2.NP 4 pole	0	1			1	25/01/2022 09:00	25/01/2022 17:00	Construct	Sets->Spiroll 2.NP 4 pole
23	1 Spiroll 2.NP 5 pole	0	1			1	26/01/2022 09:00	26/01/2022 17:00	Construct	Sets->Spiroll 2.NP 5 pole
24	1 Spiroll 2.NP 6 pole	0	1			1	27/01/2022 09:00	27/01/2022 17:00	Construct	Sets->Spiroll 2.NP 6 pole
25	1 Spiroll 2.NP 7 pole	0	1			1	28/01/2022 09:00	28/01/2022 17:00	Construct	Sets->Spiroll 2.NP 7 pole
26	1 Spiroll 2.NP 8 pole	0	1			1	31/01/2022 09:00	31/01/2022 17:00	Construct	Sets->Spiroll 2.NP 8 pole
27										

Obrázek 114 - Forma harmonogramu v softwaru Microsoft Excel

Animace výstavby se generuje automaticky z harmonogramu, který vytvoříme. Pro samotnou animaci můžeme upravovat její vlastnosti jako například přepsání data začátku a konce, rozměr intervalu a jeho jednotky nebo se dá nastavit porovnání animace výstavby plánované vs reálné, která má vazby právě na plánovaný a aktuální start a konec úkonu. Detailní nastavení simulace vidíme na obrázku 115.

Simulation Settings X

**Start / End Dates**

Override Start / End Dates

Start Date  
09:00:00 27/12/2021

End Date  
17:00:00 31/01/2022

**Interval Size**

5 Percent

Show all tasks in interval

**Playback Duration (Seconds)**

20

**Overlay Text**

Edit Top

**Animation**

No Link

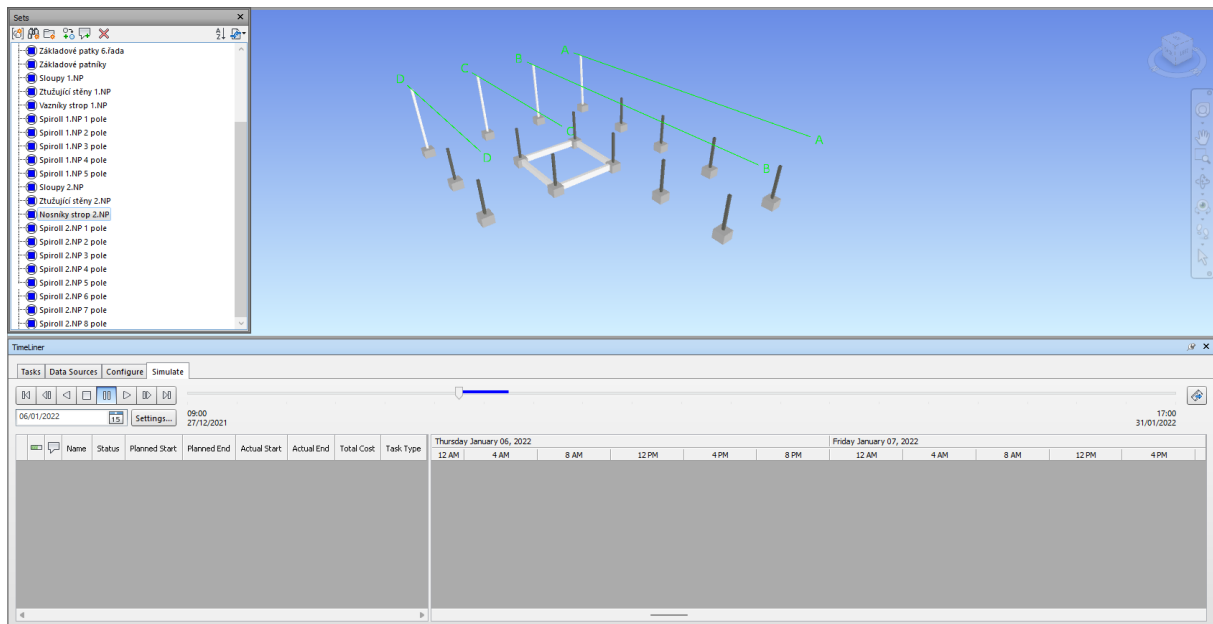
**View**

Planned  
 Planned (Actual Differences)  
 Planned against Actual  
 Actual  
 Actual (Planned Differences)

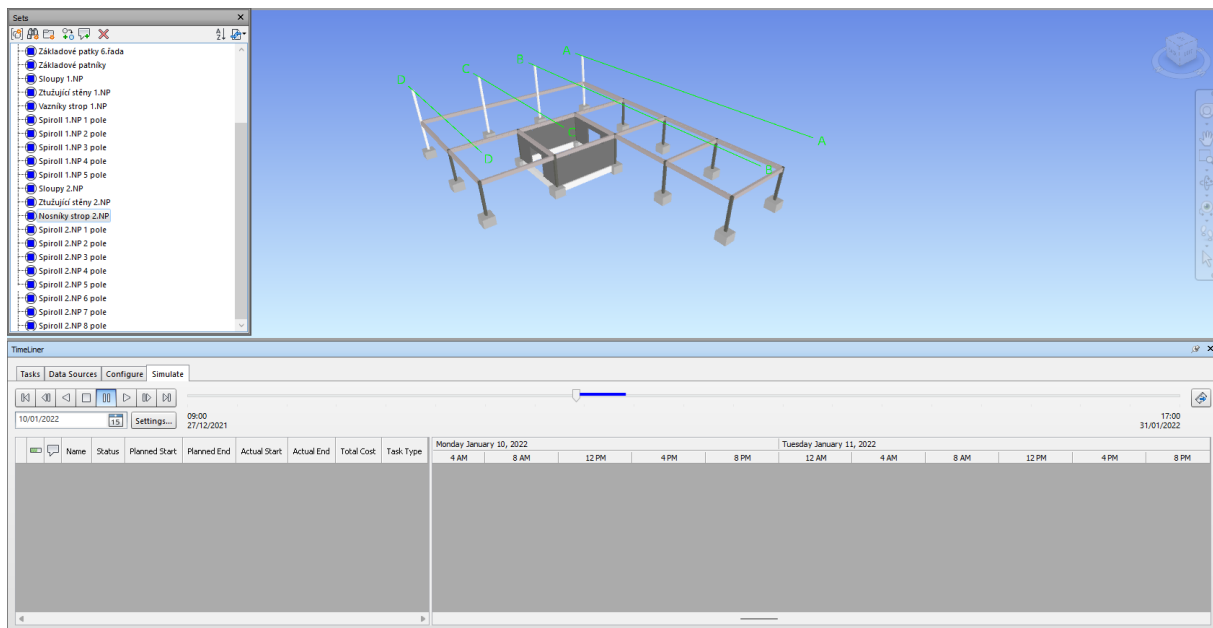
OK Cancel Help

Obrázek 115 - Nastavení simulace

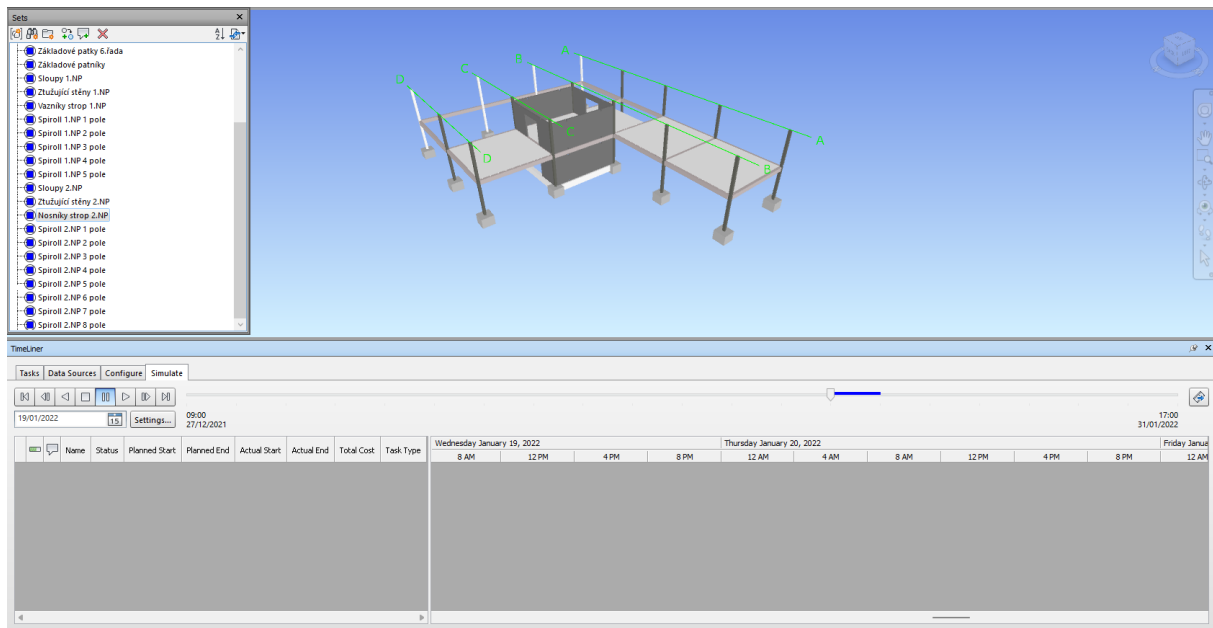
Samotná animace se dá exportovat do JPG, PNG nebo video formátu. Export do obrázkové formátu funguje jako export jednotlivých snímků, tedy sérii obrázků. Animaci ve video formátu bohužel není možné do práce vložit, a tak je znázorněna alespoň sérii obrázků viz níže.



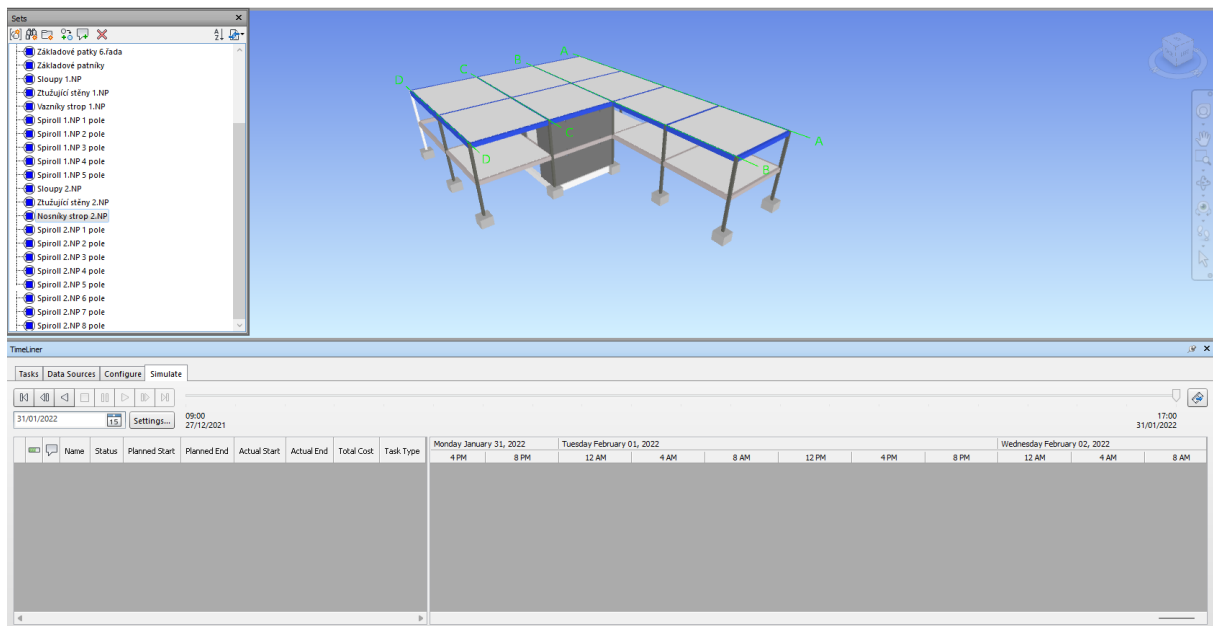
Obrázek 116 - 1. Fáze simulace



Obrázek 117 - 2. Fáze simulace



Obrázek 118 - 3. Fáze simulace



Obrázek 119 - 4. Fáze simulace



## 7. Závěr

Všechny aspekty diplomové práce, které byly na počátku stanoveny, jsem splnil. Bohužel se jednalo o práci založenou na softwaru Revit, který byl v určitých částech práce poměrně limitující, hlavně pokud se jednalo o parametrizaci prvků jako je stěna, schodiště a podobně. I přes tento fakt vznikla knihovna prvků, které jsou pro systém DPS plně funkční. Skript pro potřeby systému DPS byl vytvořen v jednoduché programovací nástavbě Dynamo. Ve výsledku byl zprovozněn do funkční podoby odpovídající požadavkům práce. Analytický model byl taktéž zachován a jeho převod mezi Robotem a Revitem funguje. Prvky jako takové fungují, chovají se dynamicky a byl parametrizován každý možný aspekt. Výsledkem z těchto prvků DPS byl vzorový model, který se nám podařilo využít k tvorbě dokumentace, pro statickou analýzu, tvorbu harmonogramu a schématu výstavby.

## 8. Literatura

Dynamo primer – návod pro programovací prostředí Dynamo

<https://primer.dynamobim.org/>

Data-Shapes – návod pro práci s custom nody Data-Shapes

<https://data-shapes.io/>

Python tutorial – návod na práci s programovacím jazykem Python

<https://docs.python.org/3/tutorial/>

LinkedIn – výukové prostředí LinkedIn

<https://www.linkedin.com/learning/me>

Koncepce Bim – implementace BIMu v rámci ČR

<https://www.koncepcebim.cz/koncepce>

czBIM – autoři klasifikačního systému SNIM

<https://snim.czbim.org/>

Prefa – výrobce prefabrikovaných železobetonových konstrukcí

<https://www.prefa.cz/>

Autodesk help – nápovědy k funkčnosti softwaru Revit

<https://help.autodesk.com/view/RVT/2022/ENU/>

## 9. Seznam použitých softwarů

1. Autodesk Revit 2022
2. Autodesk Robot Structural Analysis Professional 2022
3. Navisworks Manage 2022
4. Microsoft Word
5. Microsoft Excel

## 10. Přílohy

1. Půdorys stropu typového projektu 1.NP
2. Půdorys stropu typového projektu 2.NP
3. Katalog prvků systému DPS

## 11. Seznam obrázků

Obrázek 1 - Šablony rodin .....	14
Obrázek 2 - Typové parametry rodin .....	15
Obrázek 3 - Zatřídění rodin do kategorií .....	16
Obrázek 4 – Typy parametrů softwaru Revit .....	17
Obrázek 5 - Tvorba nového parametru .....	17
Obrázek 6 - Příklad modelovacího prostředí rodiny nosníku .....	18
Obrázek 7 - Modelovací nástroje .....	19
Obrázek 8 - Příklad použití referenčních rovin .....	20
Obrázek 9 - Typický model nosníku .....	20
Obrázek 10 - Modelování sloupu .....	21
Obrázek 11 - Schéma parametrizace kotev .....	22
Obrázek 12 - Parametrizace kotev v rodině nosného prvku .....	23
Obrázek 13 - Vazba kotev na značení nosných prvků systému DPS .....	24
Obrázek 14 - Příklad rodiny kotvy sloup nosník .....	26
Obrázek 15 - Čelní pohled na kotvu sloup nosník .....	26
Obrázek 16 - Horní pohled na kotvu sloup nosník .....	27
Obrázek 17 - Kotva sloup sloup horní .....	27
Obrázek 18 - 3D pohled rodiny nosníku .....	28
Obrázek 19 - Ukázka parametrů rodiny nosníku .....	29
Obrázek 20 - Půdorysný pohled na rodinu nosníku .....	30
Obrázek 21 - Nosník systému DPS v modelu .....	30
Obrázek 22 - Nosník systému DPS prodloužený .....	31
Obrázek 23 - Nosník DPS před úpravou parametru kotvy .....	31
Obrázek 24 - Nosník DPS po úpravě parametru kotvy .....	32
Obrázek 25 - Parametry kotvy nosník sloup .....	33
Obrázek 26 - Chyba při tvorbě dutého tvaru s vazbou na rozměry kotvy .....	34
Obrázek 27 - Tvorba dutého tvaru pro ořez v místě kotev pro nosník Spiroll .....	34
Obrázek 28 - Omezení použití parametrů v rovnici .....	35
Obrázek 29 - Rodina sloupu přes jedno patro .....	37
Obrázek 30 - Půdorysný pohled na rodinu sloupu .....	37
Obrázek 31 - Kotva sloup nosník a přiřazení parametru v nosníku .....	38
Obrázek 32 - Parametr viditelnosti pro kotvy sloup nosník .....	39
Obrázek 33 - Parametrizace pozice kotvy sloup nosník .....	39
Obrázek 34 - Vazba kotvy sloup nosník a nosník sloup .....	40
Obrázek 35 - Pohled na sloup přes dvě patra .....	41
Obrázek 36 - Parametrizace dutého tvaru ořezání v patě sloupu .....	42
Obrázek 37 - Parametrizace dutého tvaru ořezání v patě sloupu půdorys .....	42
Obrázek 38 - Výřez v místě kotvy i přes její vypnutou viditelnost .....	43
Obrázek 39 - Viditelnost kotvy sloup sloup horní .....	44
Obrázek 40 - Řešení problému výřezu .....	44
Obrázek 41 - 3D pohled na sloup a vazby k jeho ořezání .....	44

Obrázek 42 - Rozdíl mezi sloupem přes jedno a přes dvě patra .....	45
Obrázek 43 - Parametrizace sloupu ve tvaru šestihranu .....	46
Obrázek 44 - Vazba kotev na sloupu ve tvaru šestihranu .....	47
Obrázek 45 - Parametrizace sloupu ve tvaru osmihranu .....	47
Obrázek 46 - Představení rodiny sloupu ve tvaru šestihranu .....	48
Obrázek 47 - Rozdíl mezi analytickým modelem nosníku a stěny .....	49
Obrázek 48 - Rodina výřezu pro stěnu .....	50
Obrázek 49 - Půdorysný pohled na rodinu ztužující stěny DPS.....	51
Obrázek 50 - Parametry ztužující stěny DPS .....	51
Obrázek 51 - Boční pohled na rodinu ztužující stěnu DPS .....	52
Obrázek 52 - Čelní pohled na rodinu ztužující stěny DPS.....	53
Obrázek 53 - 3D pohled na ztužující stěnu DPS .....	54
Obrázek 54 - Čelní pohled na rodinu Spiroll panelu .....	55
Obrázek 55 - Parametrizace dutého tvaru ořezání panelu Spiroll .....	56
Obrázek 56 - Půdorysný pohled rodiny Spiroll panelu.....	56
Obrázek 57 - Čelní pohled na rodinu základového prahu.....	57
Obrázek 58 - Půdorysný pohled na rodinu základového patníku .....	58
Obrázek 59 - Schéma původního záměru funkčnosti pluginu .....	60
Obrázek 60 - Schéma funkčnosti skriptu.....	61
Obrázek 61 - Celkový přehled skriptu pro vkládání prvků DPS.....	61
Obrázek 62 - 1. část skriptu systému DPS .....	62
Obrázek 63 - 1. vyskakovací okno procesu kopírování prvků .....	62
Obrázek 64 - 2. Část skriptu systému DPS.....	63
Obrázek 65 - 3. Část skriptu systému DPS.....	64
Obrázek 66 - 2. Vyskakovací okno pro výběr kategorie rodiny.....	64
Obrázek 67 - 4. Část skriptu systému DPS.....	65
Obrázek 68 - 5. Část skriptu systému DPS.....	66
Obrázek 69 - 3. Vyskakovací okno s výběrem rodiny.....	66
Obrázek 70 - 6. Část skriptu systému DPS.....	67
Obrázek 71 - 7. Část skriptu systému DPS.....	67
Obrázek 72 – Víceúrovňový seznam .....	68
Obrázek 73 - 8. Část skriptu systému DPS.....	69
Obrázek 74 - 9. Část skriptu systému DPS.....	70
Obrázek 75 - 10. Část skriptu systému DPS.....	70
Obrázek 76 - 4. Vyskakovací okno pro zadání hodnot parametrům vložené rodiny .....	71
Obrázek 77 - 11. Část skriptu systému DPS.....	72
Obrázek 78 - 5. Vyskakovací okno zadání názvu nového typu.....	72
Obrázek 79 - Vložený prvek vygenerovaný skriptem.....	73
Obrázek 80 - Dostupné rodiny ve zdrojovém souboru .....	74
Obrázek 81 - 3D zobrazení dostupných rodin ve zdrojovém souboru.....	75
Obrázek 82 - Dynamo player.....	75
Obrázek 83 - Značení prvků DPS ve vzorových modelech .....	78
Obrázek 84 - 3D pohled na model č.1 .....	78
Obrázek 85 - Detail vazby prvků systému DPS.....	79

Obrázek 86 - Detail vazby prvků systému DPS 2 .....	79
Obrázek 87 - Export modelu rámu do IFC .....	80
Obrázek 88 - Grafický x analytický model příčného rámu .....	81
Obrázek 89 - Import IFC rámu do softwaru Scia Engineer .....	82
Obrázek 90 - Vazby rámu v softwaru Scia Engineer .....	82
Obrázek 91 - Stěna grafický x analytický model .....	83
Obrázek 92 - Analytický model vzorového modelu č.1 .....	84
Obrázek 93 - Export modelu do Robot Structural Analysis .....	84
Obrázek 94 - Nastavení exportu do Robot .....	85
Obrázek 95 - Příprava profilů v Robotovi pro import modelu .....	86
Obrázek 96 - Přiřazení profilu referenčnímu prvku DPS .....	86
Obrázek 97 - Analytický model v Robotu .....	87
Obrázek 98 - Grafický model v Robotu .....	87
Obrázek 99 - Výpočet vnitřních sil od vlastní tíhy .....	88
Obrázek 100 - 3D model vzorového modelu č.2 .....	89
Obrázek 101 - Značení prvků DPS ve výkresové dokumentaci .....	90
Obrázek 102 - Příklad značení prvků DPS .....	90
Obrázek 103 - Detail rodiny multikategorie .....	91
Obrázek 104 - Výkaz prvků z modelu č.1 .....	92
Obrázek 105 - Výkaz prvků z modelu č.2 .....	93
Obrázek 106 - Porovnání prvků z modelu 1 a modelu 2 .....	94
Obrázek 107 - Tvorba výkazu prvků DPS .....	95
Obrázek 108 - Kombinované parametry .....	96
Obrázek 109 - Parametry kotvy sloup nosník .....	97
Obrázek 110 - 3D model vzorového modelu č.1 v softwaru Navisworks .....	98
Obrázek 111 - Vytvořené sety modelu č.1 .....	99
Obrázek 112 - Automatická tvorba harmonogramu .....	100
Obrázek 113 - Export harmonogramu do .CSV souboru .....	100
Obrázek 114 - Forma harmonogramu v softwaru Microsoft Excel .....	101
Obrázek 115 - Nastavení simulace .....	102
Obrázek 116 - 1. Fáze simulace .....	103
Obrázek 117 - 2. Fáze simulace .....	103
Obrázek 118 - 3. Fáze simulace .....	104
Obrázek 119 - 4. Fáze simulace .....	104