Optimization of material usage

Harald Lyngbye, s203615

Thomas Lyskjær, s203626

Kasper Vang Bengtsson, s203628

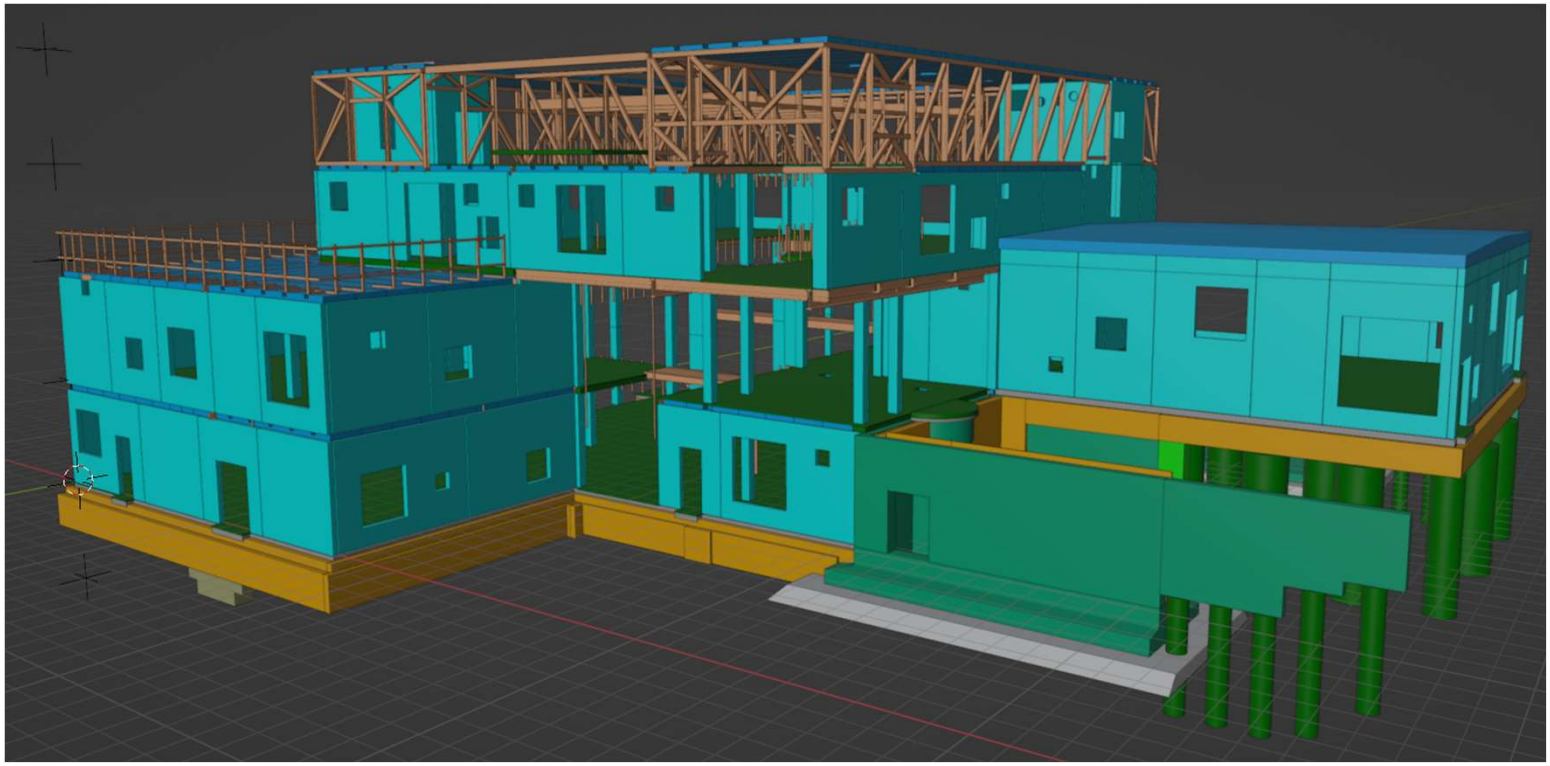
Goal

User

Process

Coding

BLENDER BIM



Goal

User

Process

Coding

- Optimizing Construction Material via FEM Analysis for Efficiency
- Reduce time

Goal

User

Process

Coding

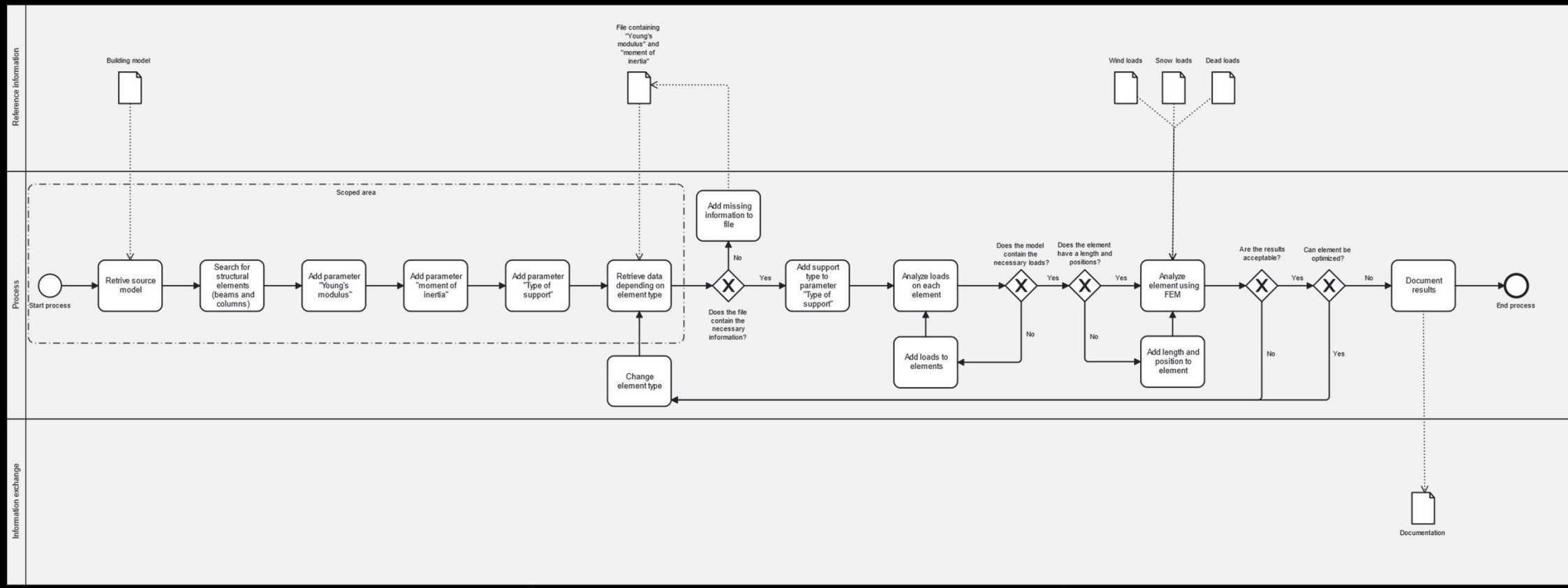
User of the script:

- Structural Engineer

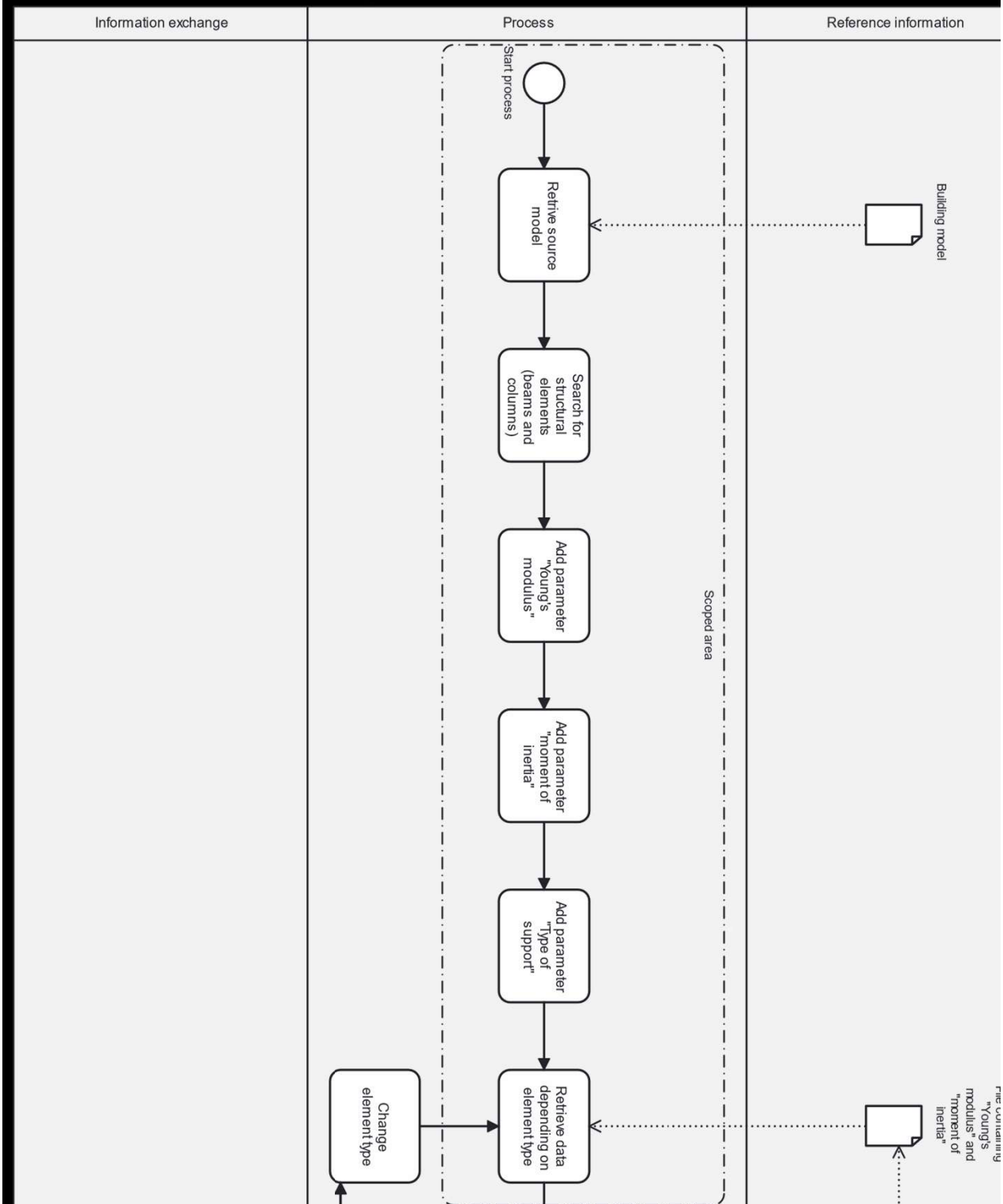
Role level:

- Analyst level 3 – ifcOpenShell

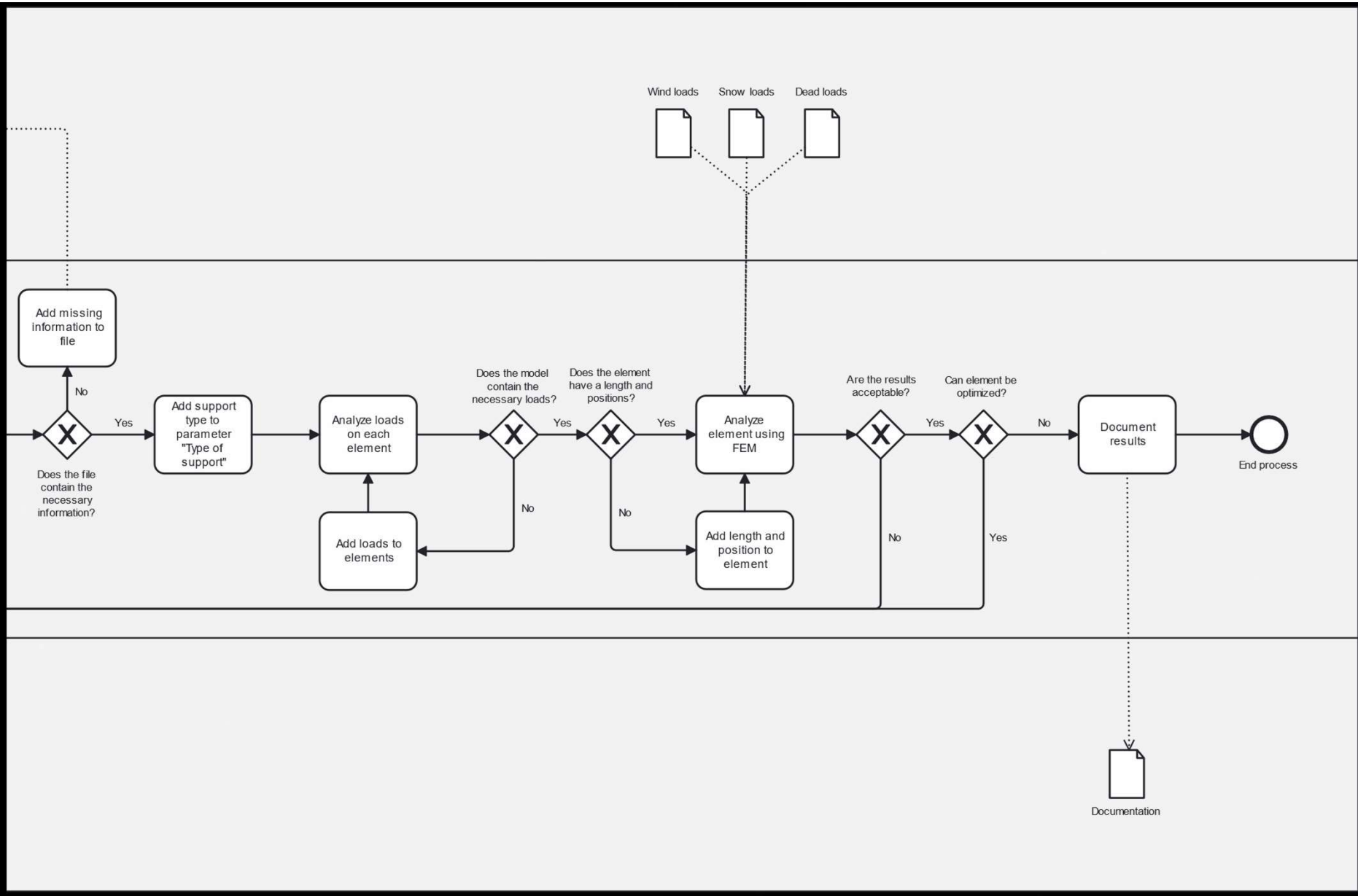
Goal User **Process** Coding



Our implementation



Future implementation



Retrieve model and configurate parameters.

Find elements in our case we use a beam.

Create new property set for the beam If none existing.

Insert or update properties and save new IFC file

```python
import ifcopenshell

# Configuration Parameters
ifc_file_path = 'C:/Users/kaspervang/Desktop/overgangs semester/Advance BIM/A2/LLYN - STRU.ifc
output_ifc_file_path = 'C:/Users/kaspervang/Desktop/overgangs semester/Advance BIM/A3/LLYN - S
global_id = '0PBox$QKjEKQwoUJvZkwUp'
property_set_name = 'Pset_FEMAnalysis'
properties_to_update = {
    'Moment of inertia, M_y': 1370000,
    'Moment of inertia, M_z': 1370000,
    'Youngs modulus, E': 2100000000,
}

# Open the ifc file
ifc_file = ifcopenshell.open(ifc_file_path)
```

Retrieve model and configurate parameters.

Find elements in our case we use a beam.

Create new property set for the beam If none existing.

Insert or update properties and save new IFC file

```
17 # Get the element by its global ID
18 element = ifc_file.by_guid(global_id)
19
20 # Check if the element is found
21 if not element:
22     print("Element not found")
23     exit()
24
25
```

Retrieve model and configurate parameters.

Find elements in our case we use a beam.

Create new property set for the beam If none existing.

Insert or update properties and save new IFC file

```python
# Function to get or create property set
def get_or_create_pset(element, pset_name):

    # Find all definition associated with the element
    for definition in element.IsDefinedBy:
        # Check if it is a property set
        if definition.is_a('IfcRelDefinesByProperties'):
            # Get the property set by function in ifcopenshell
            property_set = definition.RelatingPropertyDefinition
            # Check if it is the same as the one we are trying to create
            if property_set.Name == pset_name:
                return property_set

    # If there a no property set with pset_name in the element
    # Creating a new property set and assigning it a unique ID and owner history
    property_set = ifc_file.createIfcPropertySet(
        GlobalId=ifcopenshell.guid.new(),
        OwnerHistory=ifc_file.by_type('IfcOwnerHistory')[0],
        Name=pset_name,
        HasProperties=[]
    )

    # Link the property set to the element
    ifc_file.createIfcRelDefinesByProperties(
        GlobalId=ifcopenshell.guid.new(),
        OwnerHistory=ifc_file.by_type('IfcOwnerHistory')[0],
        RelatedObjects=[element],
        RelatingPropertyDefinition=property_set
    )
    return property_set
```

Goal User Process **Coding**

Retrieve model and configurate parameters.

Find elements in our case we use a beam.

Create new property set for the beam If none existing.

Insert or update properties and save new IFC file

```
try:
    property_set = get_or_create_pset(element, property_set_name)

    for property_name, property_value in properties_to_update.items():

        # check if the property already exisist in the property set
        existing_property = next(
            (prop for prop in property_set.HasProperties if prop.Name == property_name),
            None
        )
        if existing_property:
            # If the property exists, update its value
            existing_property.NominalValue.wrappedValue = property_value
        else:
            # Make new property and add to property set
            nominal_value = ifc_file.createIfcReal(property_value)
            new_property = ifc_file.createIfcPropertySingleValue(
                Name=property_name,
                NominalValue=nominal_value,
            )

            # add the new property to the list and update
            properties_list = list(property_set.HasProperties) if isinstance(property_set.HasP
            properties_list.append(new_property)
            property_set.HasProperties = properties_list
    # Save the IFC file with the new properties
    ifc_file.write(output_ifc_file_path)
# Handling exceptions and printing an error message.
except Exception as general_exception:
    print(f"An error occurred: {general_exception}")
```