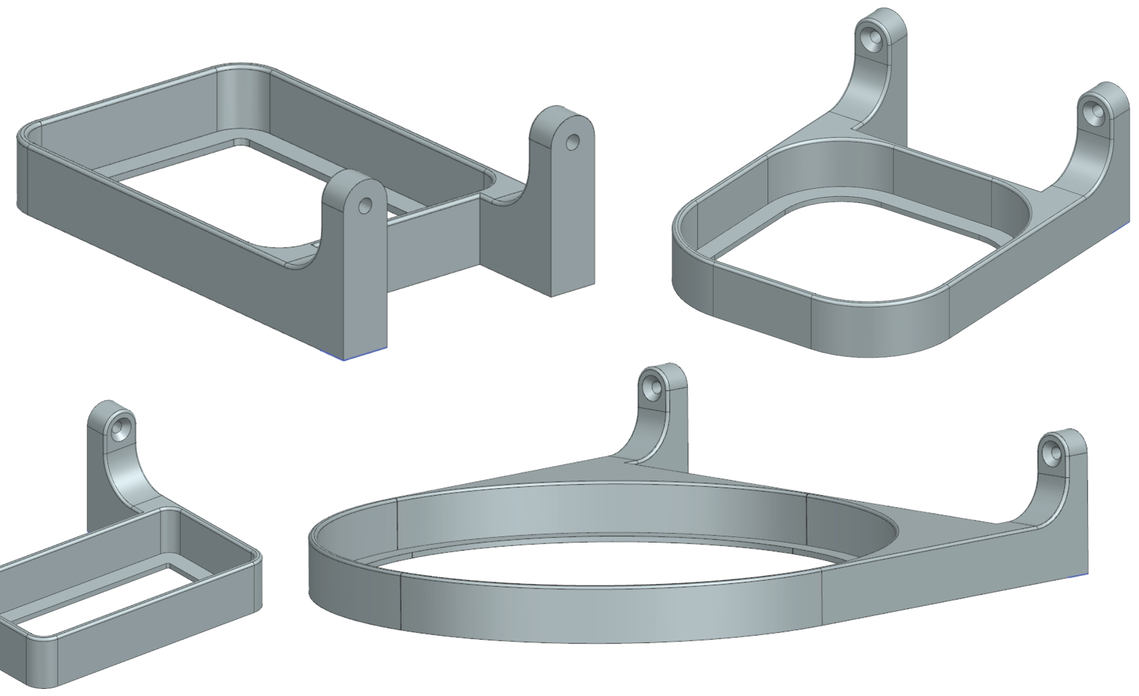
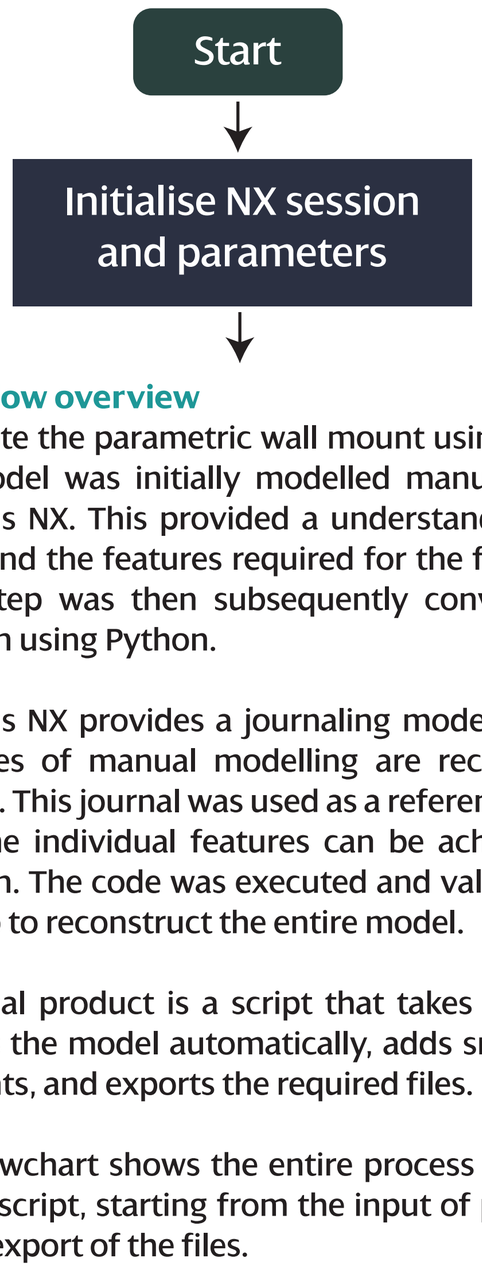
Parametric wall mount modeling using NXOpen



Introduction

This project looks at the possibility of using NXOpen to build a parametric model of a wall mount. Instead of modelling every feature manually, the entire model will be built using a Python script. The user just needs to give some basic parameters, including product size and shape. All modelling steps and exportation of the model are done by the script.

The primary reason for this project is to be able to quickly generate a lot of different wall mounts for devices such as speakers, routers, or any other device. The process is automated, making it simple and quicker to design and modify customized mounts according to any product size.



Workflow overview

To create the parametric wall mount using NXOpen, the model was initially modelled manually within Siemens NX. This provided a understanding of the steps and the features required for the final model. Each step was then subsequently converted into NXOpen using Python.

Siemens NX provides a journaling mode where the activities of manual modelling are recorded in a journal. This journal was used as a reference point of how the individual features can be achieved with NXOpen. The code was executed and validated step by step to reconstruct the entire model.

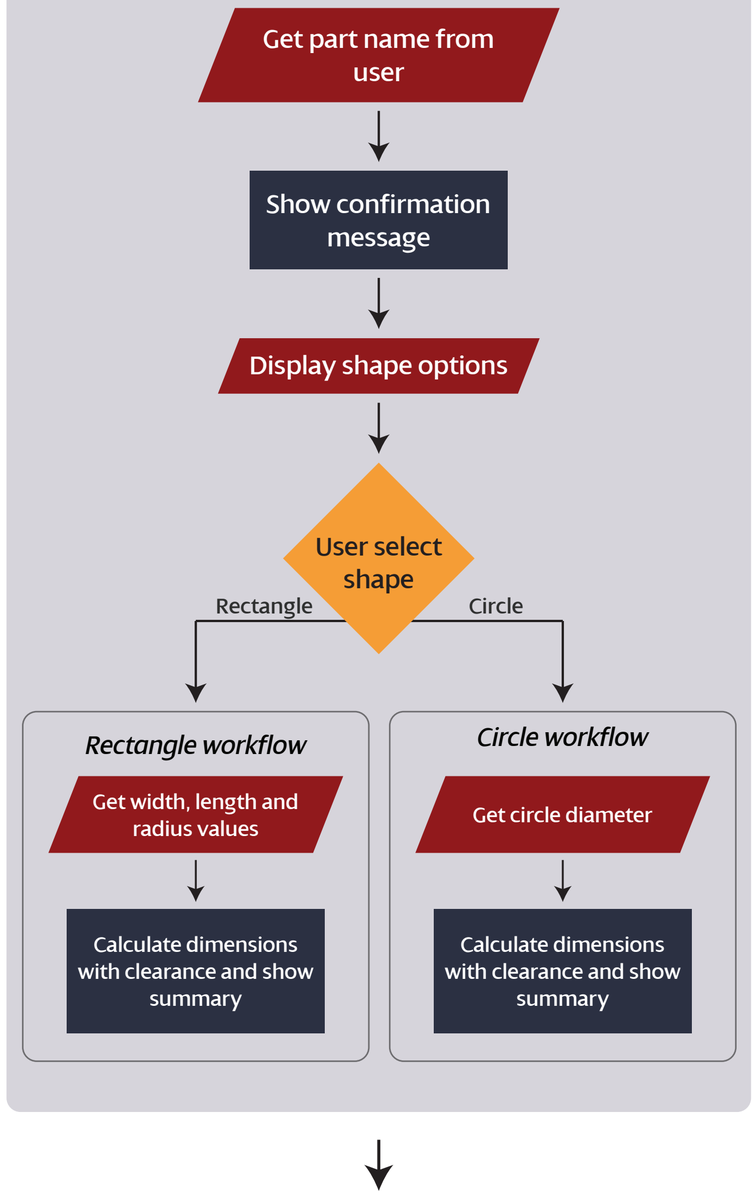
The final product is a script that takes user input, creates the model automatically, adds small design elements, and exports the required files.

The flowchart shows the entire process carried out by the script, starting from the input of parameters to the export of the files.

Parametic customization

The script is designed to generate different wall mounts based on a few user-defined parameters. After launching the script, a dialog asks the user to input the part name, choose a shape (rectangle or circle), and enter dimensions such as width, length, and radius or just the diameter if the circle option was chosen. The script automatically applies a clearance to these dimensions to make sure the product fits properly.

Based on the selected shape and size, all other parts of the wall mount, like the walls, floor, and mounting parts, are created with adjusted positions and sizes. This makes it easy to recreate the model for different devices without having to redo the design manually.



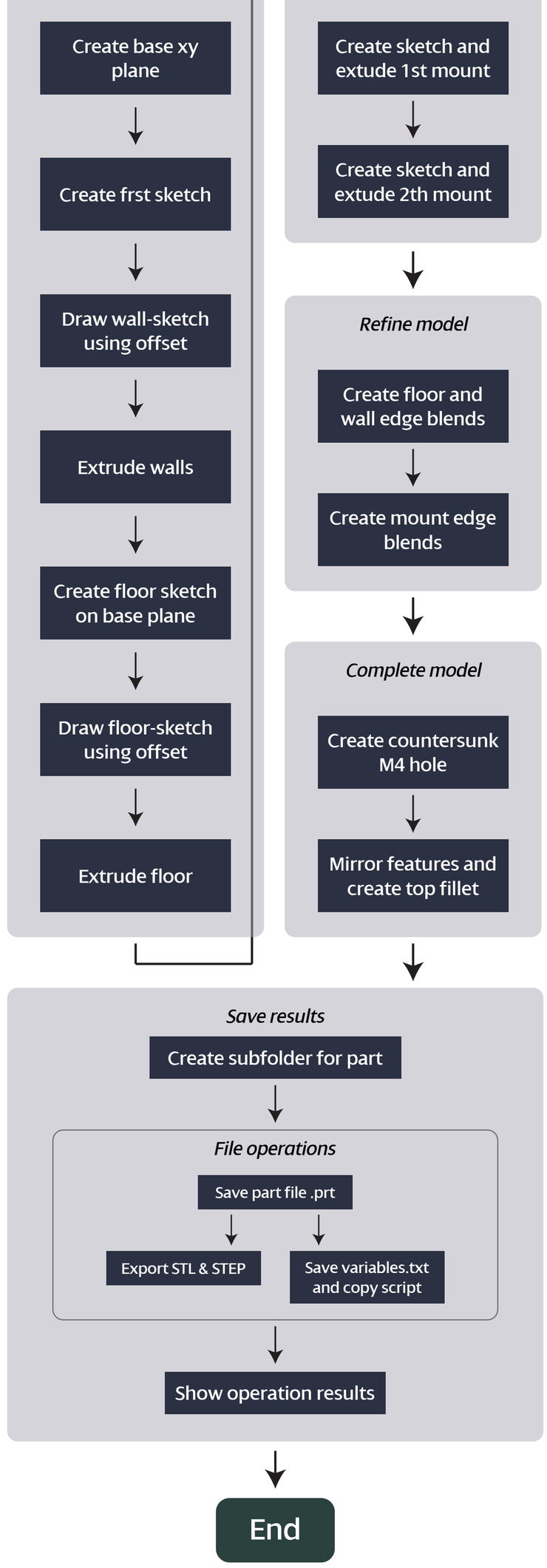
Other parameters are defined by default, but can easily be changed in the script:

```
folder = "C:\\Users\\Dennis\\Documents\\models"
clearance = 0.20              # Clearance for the width/length
height = 20.0                 # Heigh of the walls
wall_thickness = 3.0          # Wall thickness
floor_thickness = 2.0         # Floor thickness
floor_width = 10.0            # Floor width
uitsteeksel_width = 20.0      # Width of the protruding cube
uitsteeksel_length = 15.0     # Length of the protruding cube
uitsteeksel_thickness = 10.0  # Thickness of the second thing
uitsteeksel_height = 50.0     # Height of the protruding cube
uitsteeksel_fillet = 15.0     # Fillet of the screw cube
top_fillet = 1                # Fillet of the entire top surface
```

Reflection & future improvement

One of the things learned during this project is the importance of working in a structured way when writing NXOpen code. At first, all code was written in one long block, which made it hard to read and adjust. Later on, the code was split into separate functions, which helped make the script much clearer and easier to manage. This also made testing and improving individual steps much more practical. Even though the main goal was reached, there is still room to improve the script by adding more features, making the code easier to reuse, or turning it into a more user-friendly tool.

DEMO



Real-life use case

In this case, the script was used to quickly generate simple wall mounts for certain devices. However, NXOpen has many other uses in real-life product development. It can be used to automate repetitive tasks, set up batch processes, or connect NX to external tools and databases. It is also useful for creating custom applications that make working in NX faster and more efficient.

Dennis Klappe
s2400758