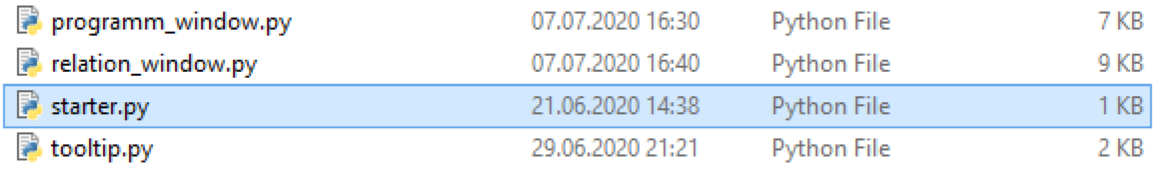
VoxML Annotator - short documentation

1. Starting the annotator:

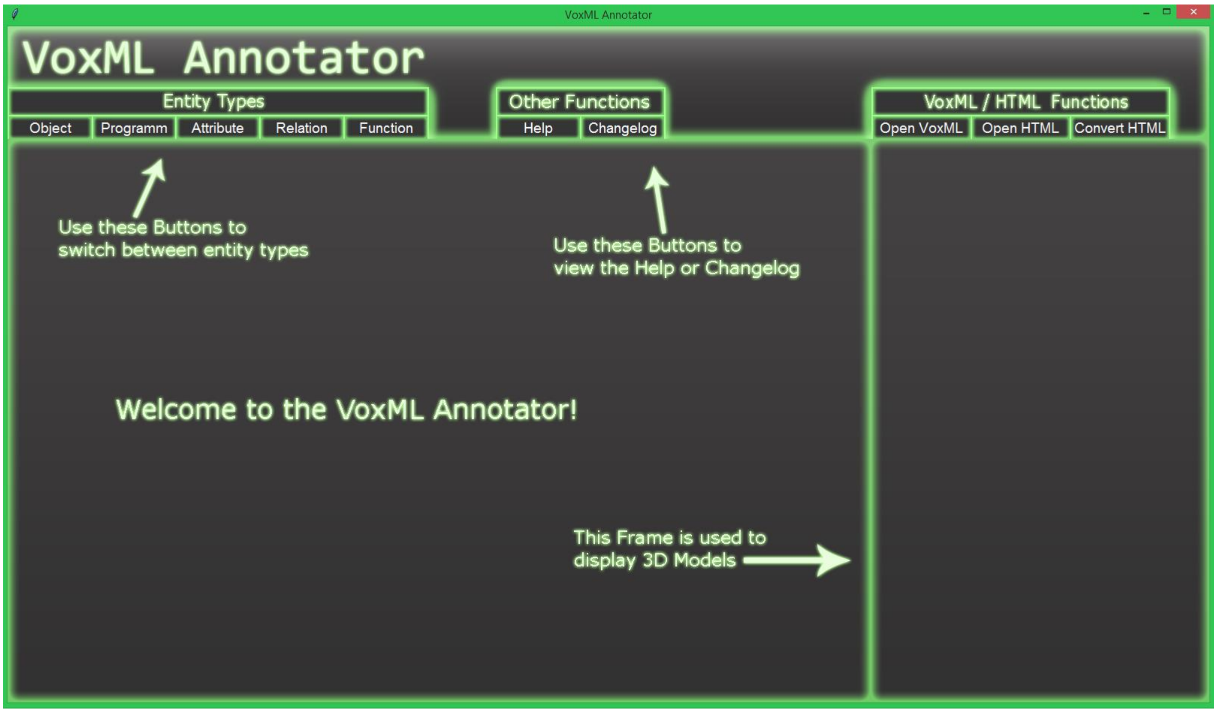
To use this programm, you need to install Python. If Python is already installed on your machine, simply execute „starter.py“ from the programm folder to start the VoxML annotator:



programm_window.py	07.07.2020 16:30	Python File	7 KB
relation_window.py	07.07.2020 16:40	Python File	9 KB
starter.py	21.06.2020 14:38	Python File	1 KB
tooltip.py	29.06.2020 21:21	Python File	2 KB

Example 1.1: starting the annotator

Once you installed all the dependencies and start the annotator, you are greeted by the welcome screen. Its supposed to give you a short introduction to the UI.



Use these Buttons to switch between entity types

Use these Buttons to view the Help or Changelog

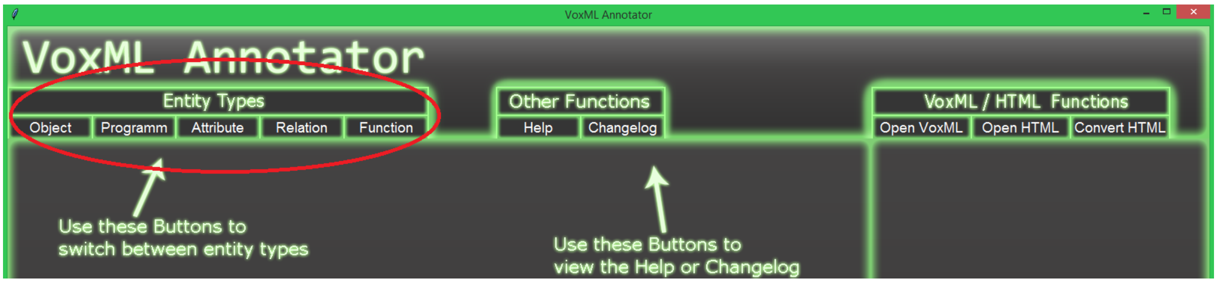
Welcome to the VoxML Annotator!

This Frame is used to display 3D Models

Example 1.2: welcome screen

2. Functions of the Annotator:

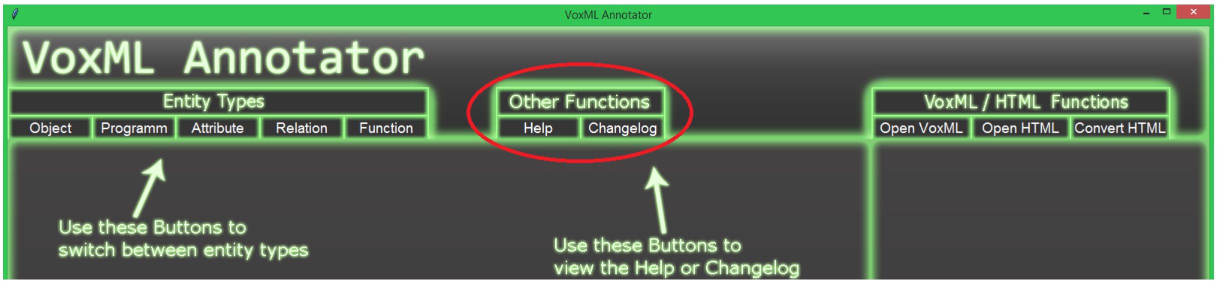
The functions of the annotator are organized in frames within the main window. To take a frame into focus and start using its functions, just use the corresponding button at the top of the annotators main window:



Use these Buttons to switch between entity types

Use these Buttons to view the Help or Changelog

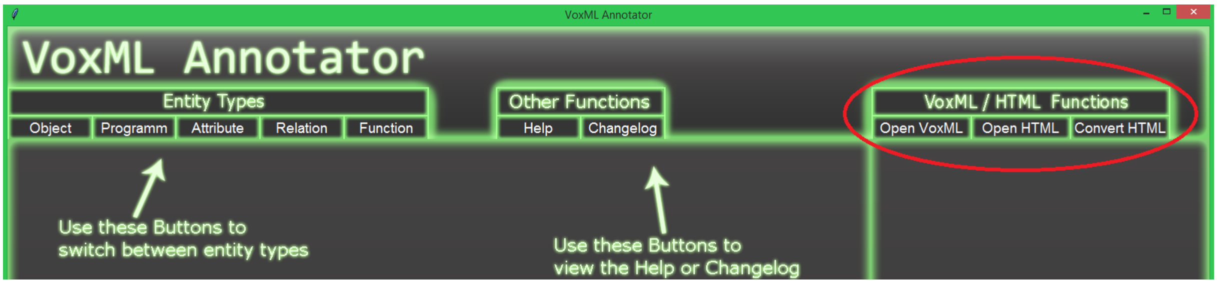
Example 2.1: selecting the entity type



Use these Buttons to switch between entity types

Use these Buttons to view the Help or Changelog

Example 2.2: opening the help / changelog



Use these Buttons to switch between entity types

Use these Buttons to view the Help or Changelog

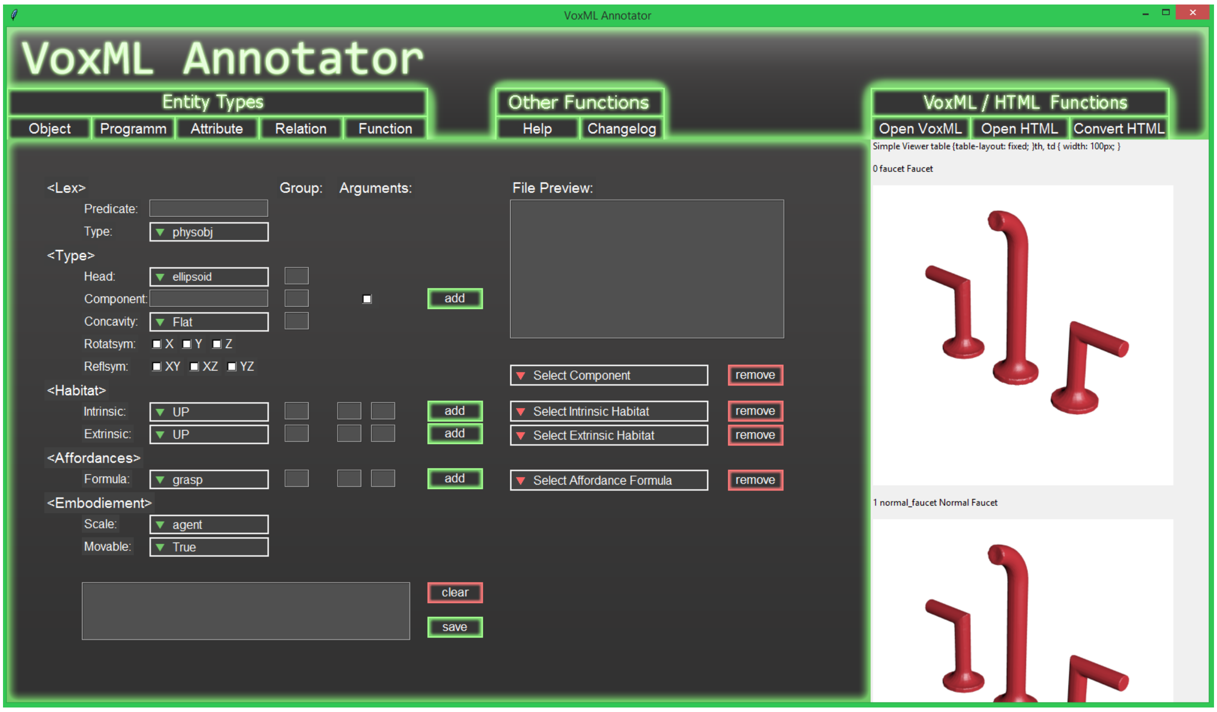
Example 2.2: using VoxML / HTML specific functions

3. Creating an annotation:

Based on your decision during the entity type selection you will see a new frame open, which contains exactly the inputs and options you need specifically for that entity type. This means that the annotation frame for objects looks differen from the one for functions. The UI is supposed to assist and accelerate the annotation process by simplifying VoxML and reducing the ammount of XML code that needs to be written.

To assist your annotation process even further you can load html files that represent some hierarchical data about an objects parts in the rightmost frame.

Before you save a file you have the option to edit it. If for example you want to delete some componets, you have added before, just select the component in the dropdown and press the remove button. Buttons that add something to the file are colored green, while buttons that remove something or reset the UI are colored in red. You can always have a look at your current files content with the file preview label.



Simple Viewer table {table-layout: fixed; }th, td { width: 100px; }

0 faucet Faucet

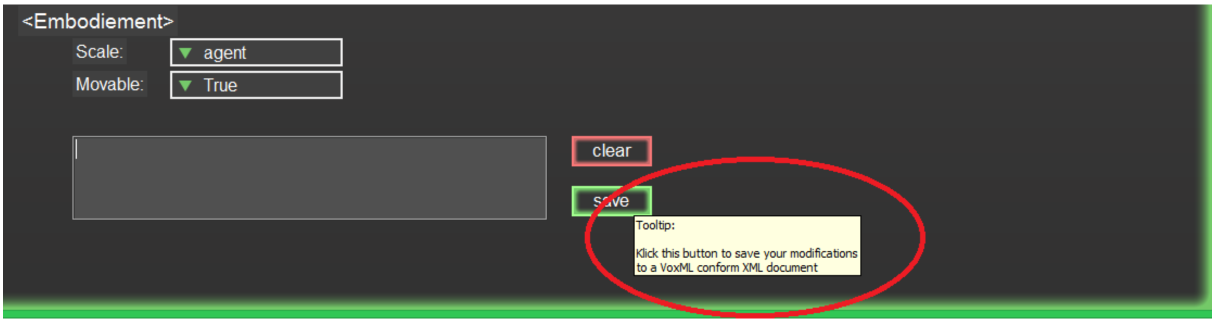
1 normal_faucet Normal Faucet

<Lex>
Predicate:
Type: physobj
<Type>
Head: ellipsoid
Component:
Concavity: Flat
Rotatsym: ■ X ■ Y ■ Z
Reflsym: ■ XY ■ XZ ■ YZ
<Habitat>
Intrinsic: UP
Extrinsic: UP
<Affordances>
Formula: grasp
<Embodiement>
Scale: agent
Movable: True

Select Component
Select Intrinsic Habitat
Select Extrinsic Habitat
Select Affordance Formula

Example 3.1: object annotator with ShapeNet model on the right side

Elements that can occur in multiple tags, like components in objects or arguments in functions are added to the document, by filling in the corresponding entry widgets, checkboxes or optionmenus and pressing the „add“ button. Once you decide to save the file, all the elements you have added before are also saved within the file. The UI only allows this for special elements where it is VoxML conform, so you dont have to worry about the VoxML syntax. Elements without an „add“ button are allowed to be filled in once, or left blank entirely. By hovering the mouse over the UI and looking at the individual tooltips, you should be able to find out which inputs are mandatory and which ones are optional.

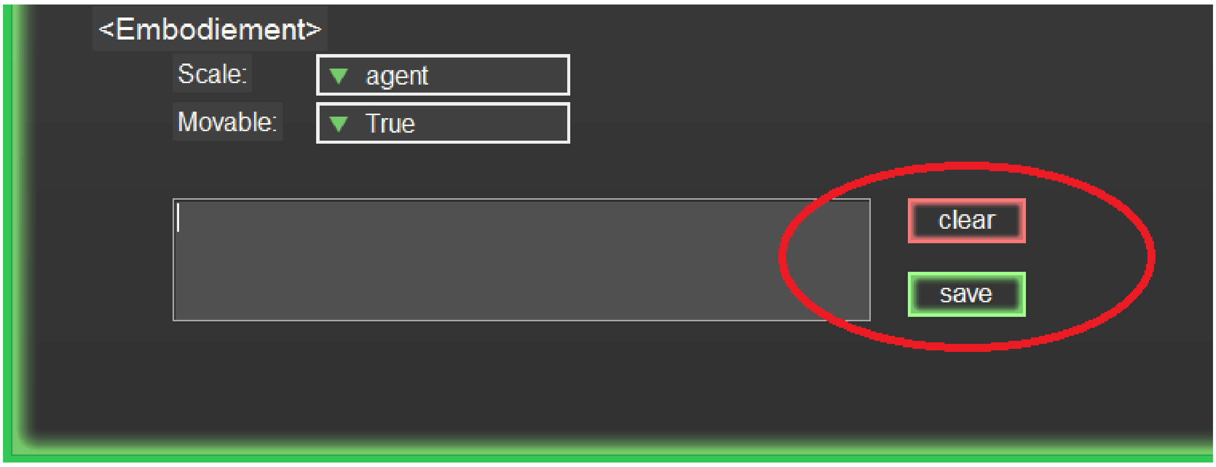


Tooltip:

Klick this button to save your modifications to a VoxML conform XML document

Example 3.2: Tooltip

If you have made a mistake during the annotation that cant simply be corrected in the UI, or you have saved your current file and want to create another one oft he same entity type, you can hit the „clear“ button, to reset the annotation frame.



Example 3.3: clear / save files

The name of a file is going to be the value you set as predicate, as this is usually what describes the content of the file in the best way.