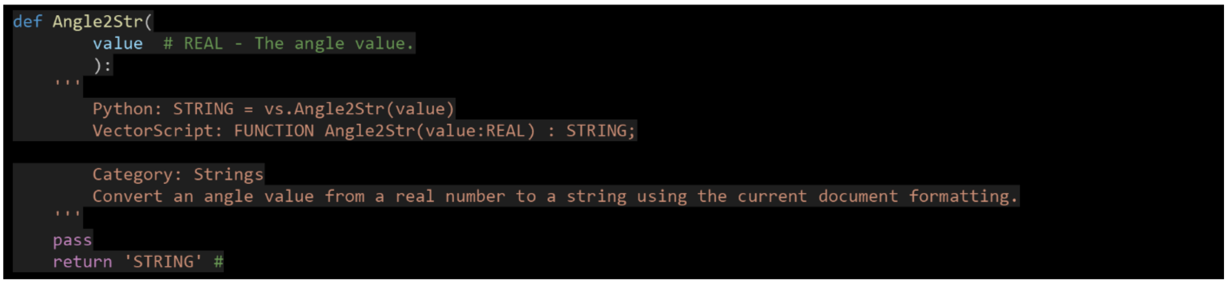
Technical University of Munich
TUM School of Engineering and Design
Chair of Computational Modeling and Simulation

Master Thesis Proposal

Using graphs to improve the interpretability of API documentation for BIM authoring software

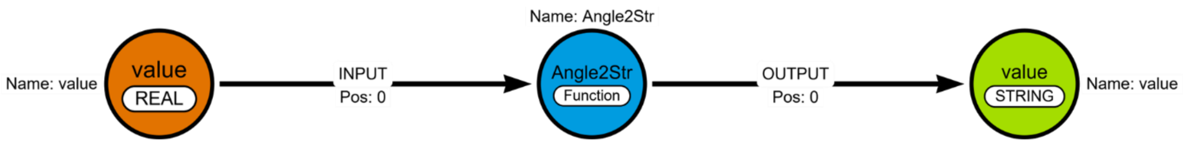
Using APIs to develop plugins for BIM authoring software is a common application in the industry. Rich APIs provided by software vendors allow developers to flexibly customize and extend the original functionalities of the software according to project requirements. To offer the greatest possible flexibility, BIM software APIs are often designed to target low-level tasks. This requires developers to consult complex API documentation to understand the functions and relationships of various APIs, thus implementing higher-level requirements. This significantly increases time costs and the learning curve.

To address this issue, the goal of this study is to create a knowledge graph for API documentation to enhance its interpretability. By integrating large language models with knowledge graphs, we aim to develop a chatbot that can intelligently assist developers with API-related questions.



```
def Angle2Str(
        value  # REAL - The angle value.
        ):
    '''
        Python: STRING = vs.Angle2Str(value)
        VectorScript: FUNCTION Angle2Str(value:REAL) : STRING;

        Category: Strings
        Convert an angle value from a real number to a string using the current document formatting.
    '''
    pass
    return 'STRING' #
```



CYPHER Statement to create this graph:

CREATE (:REAL {Name: "value"})-[:INPUT {Pos: 0}]->(:Function {Name: "Angle2Str"})-[:OUTPUT {Pos: 0}]->(:STRING {Name: "value"})

Figure 1 Possible graphical formulation for an API function of Vectorworks

Task

1. Literature review on the state-of-the-art in relevant domains
2. Implementation of a suitable workflow that includes:
    - Processing and cleaning API documentation and automatically transforming it into a suitable graph structure
    - Exploring and mining graph patterns
    - Developing a chatbot based on this knowledge graph
3. Evaluation of the proposed method

You have

1. Programming skills (Python) and motivation to work with graphs and AI.
2. As a plus:
    - Experience with Vectorworks
    - Experience with API programming
    - Experience with graph processing tools (neo4j, networkx, etc.)

Supervision

Changyu Du
changyu.du@tum.de

Sebastian Esser
sebastian.esser@tum.de