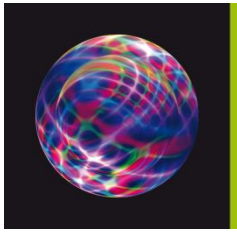
Lowering the Barrier to Entry to the UtilityNetwork ADE



EIFER

M.Sc. Isaac Boates

M.Sc. Alexandru Nichersu

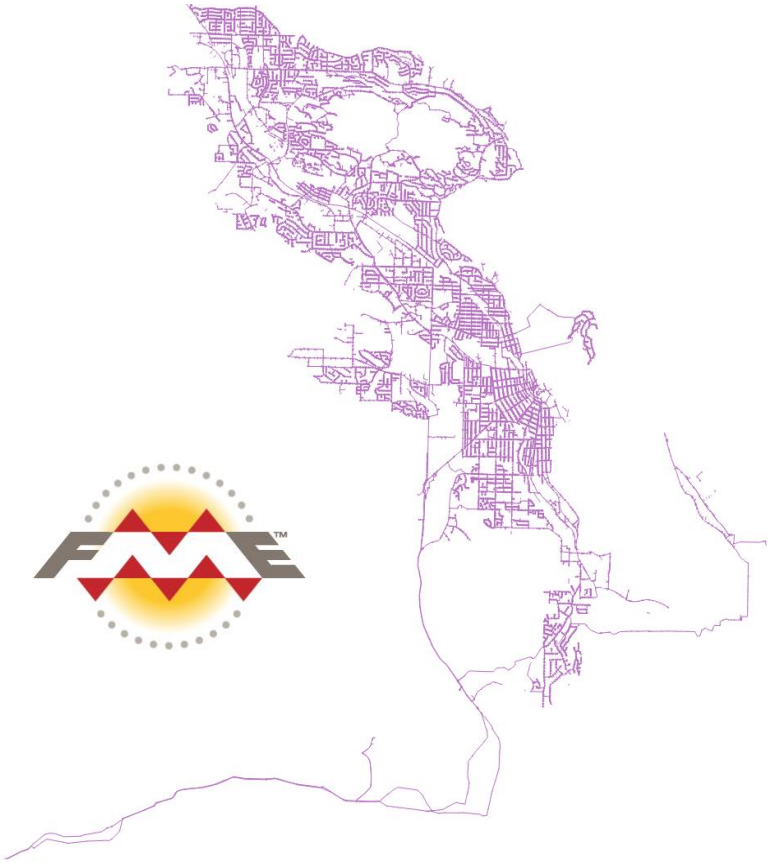
EIFER

Nanaimo Sample

- Water network released as open data in ESRI shapefile format
- Converted into first public UtilityNetwork ADE sample using FME here at TUM

- Contained:

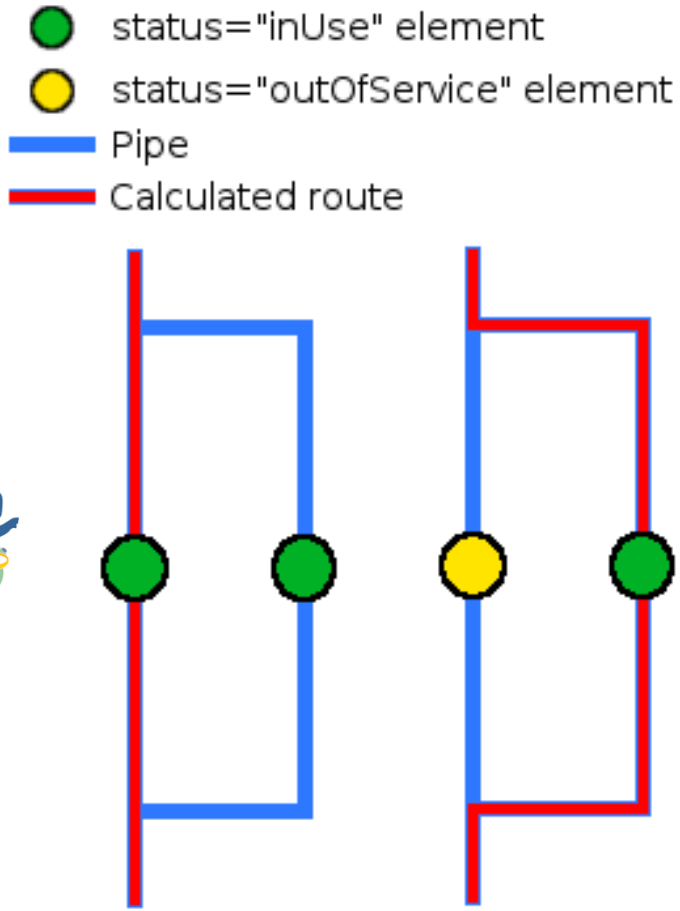
1. A Network Element
2. A NetworkGraph Element
3. LiquidMedium Element
4. ExteriorMaterial Elements
5. RoundPipe Elements
6. TerminalElements
7. Node Elements
8. InteriorFeatureLink Elements
9. FeatureGraph Elements
10. InterFeatureLink Elements



EIFER

Getting Functional

- Added “appurtenances”
  - Valves
  - Reservoirs
  - Junctions
  - Etc.
- Imported sample into 3DCityDB
- Experimented with routing between elements using pgRouting
- Found ways to simulate interruptions based on feature properties
- Wrote a paper!



ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-4, 2018
ISPRS TC IV Mid-term Symposium “3D Spatial Information Science – The Engine of Change”, 1–5 October 2018, Delft, The Netherlands

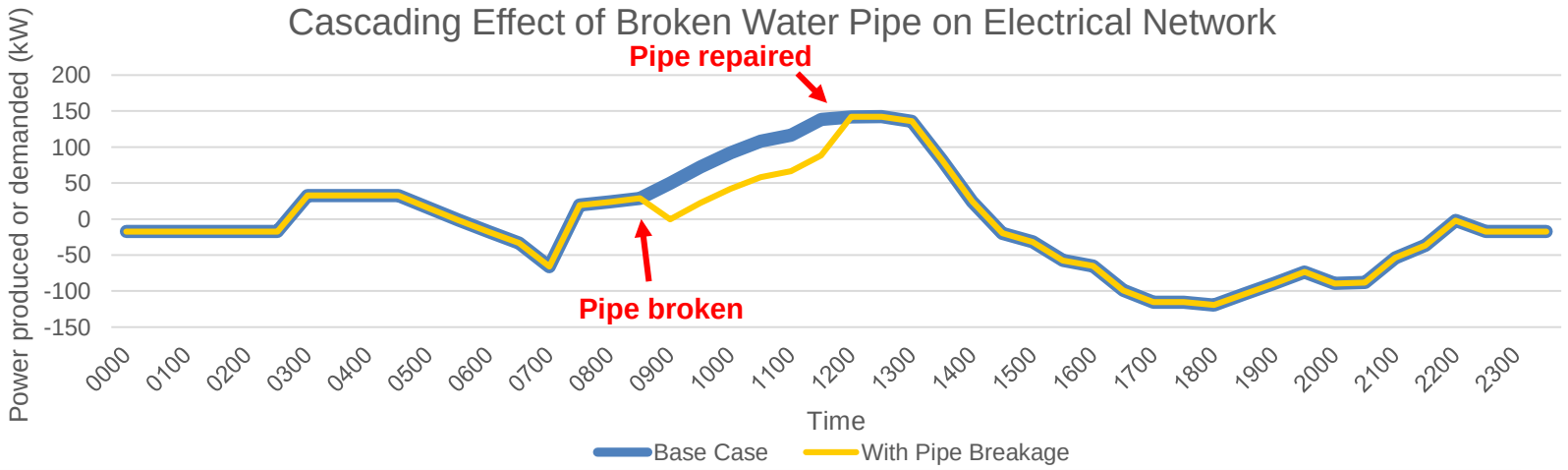
NETWORK MODELLING AND SEMANTIC 3D CITY MODELS: TESTING THE MATURITY OF THE UTILITY NETWORK ADE FOR CITYGML WITH A WATER NETWORK TEST CASE

I. Boates[1], G. Agugiaro[2], A. Nichersu[1]

EIFER

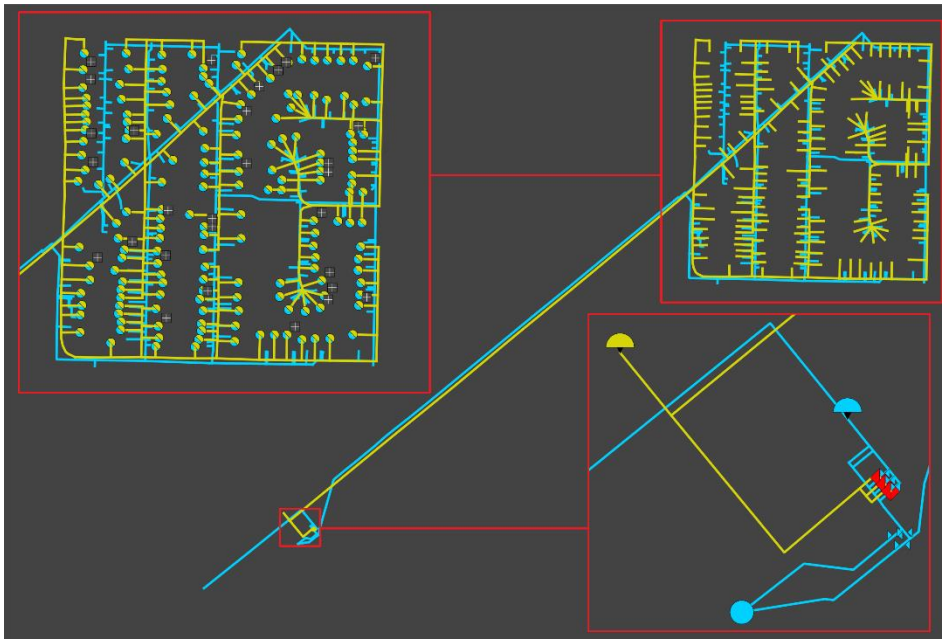
Mastering the ADE

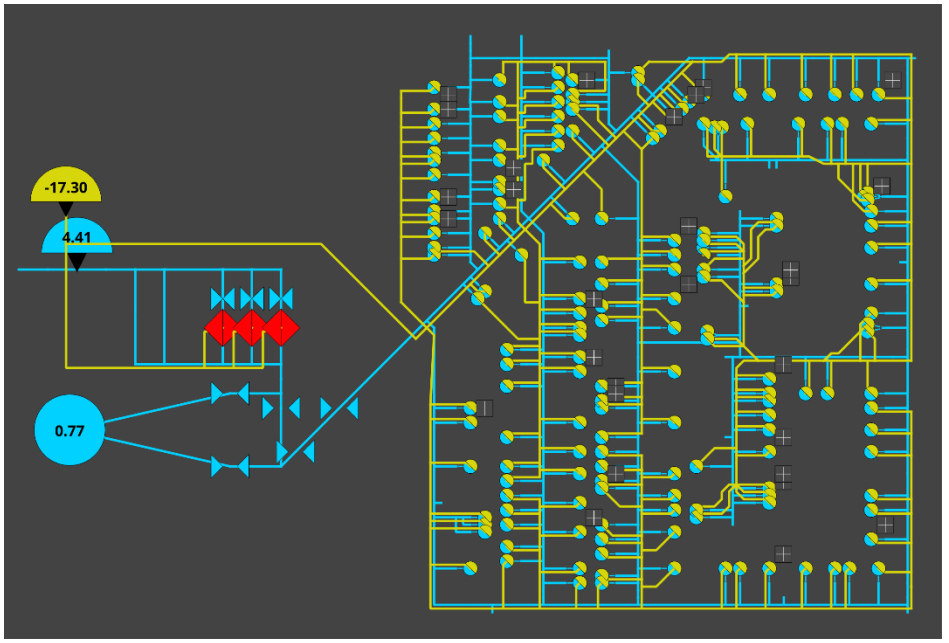
- Became the subject of master thesis
- Added an electrical network derived from the local roads
- Wrote a Python API to interface with the database
- Wrote a model to detect changes in electrical output from water flow fluctuations
- Made a second “schematic” representation



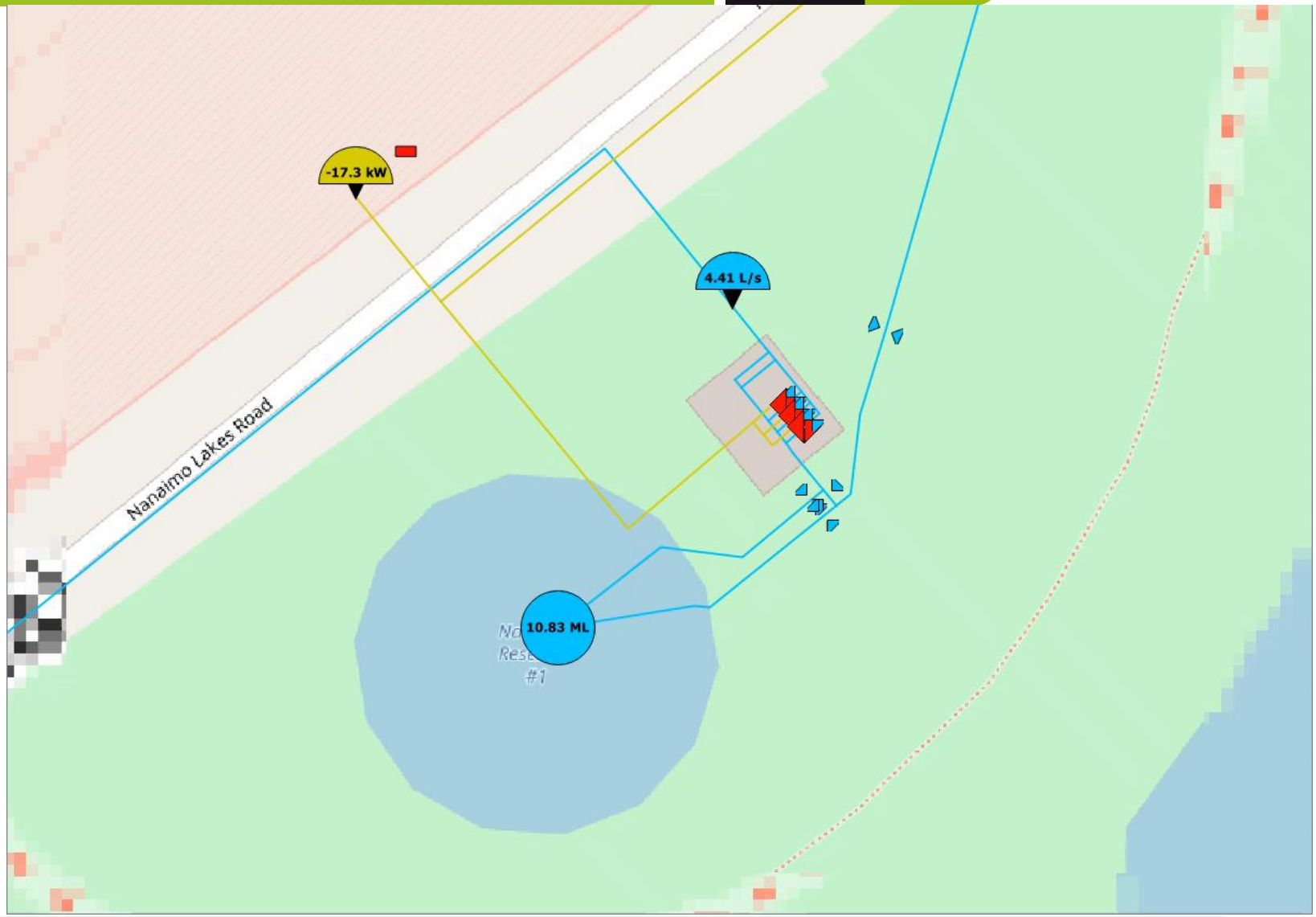
EIFER

Mastering the ADE



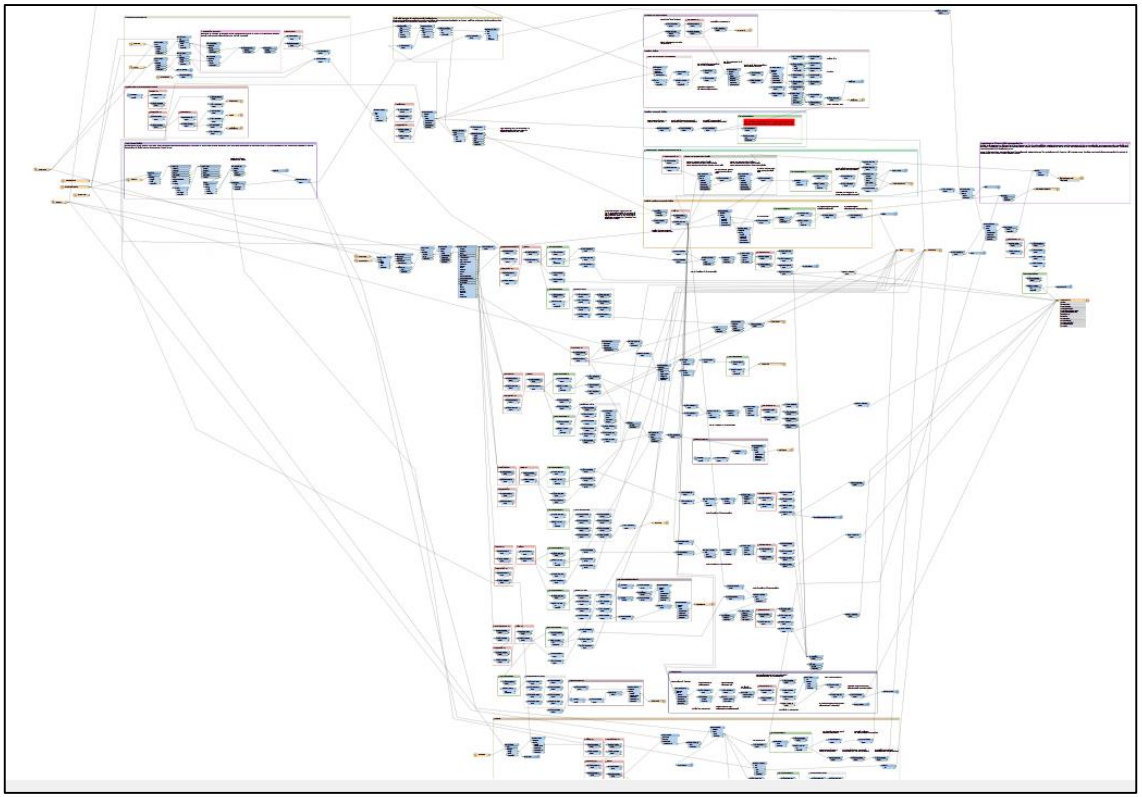


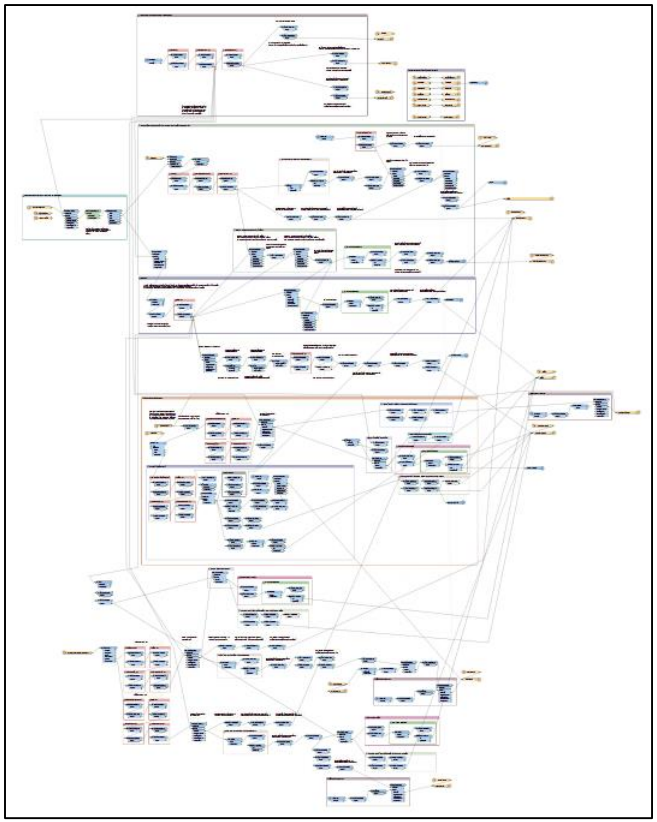
Previous Work



EIFER

The Nanaimo work has been useful for us experts, but it may be daunting to newcomers…





Time for a clean slate

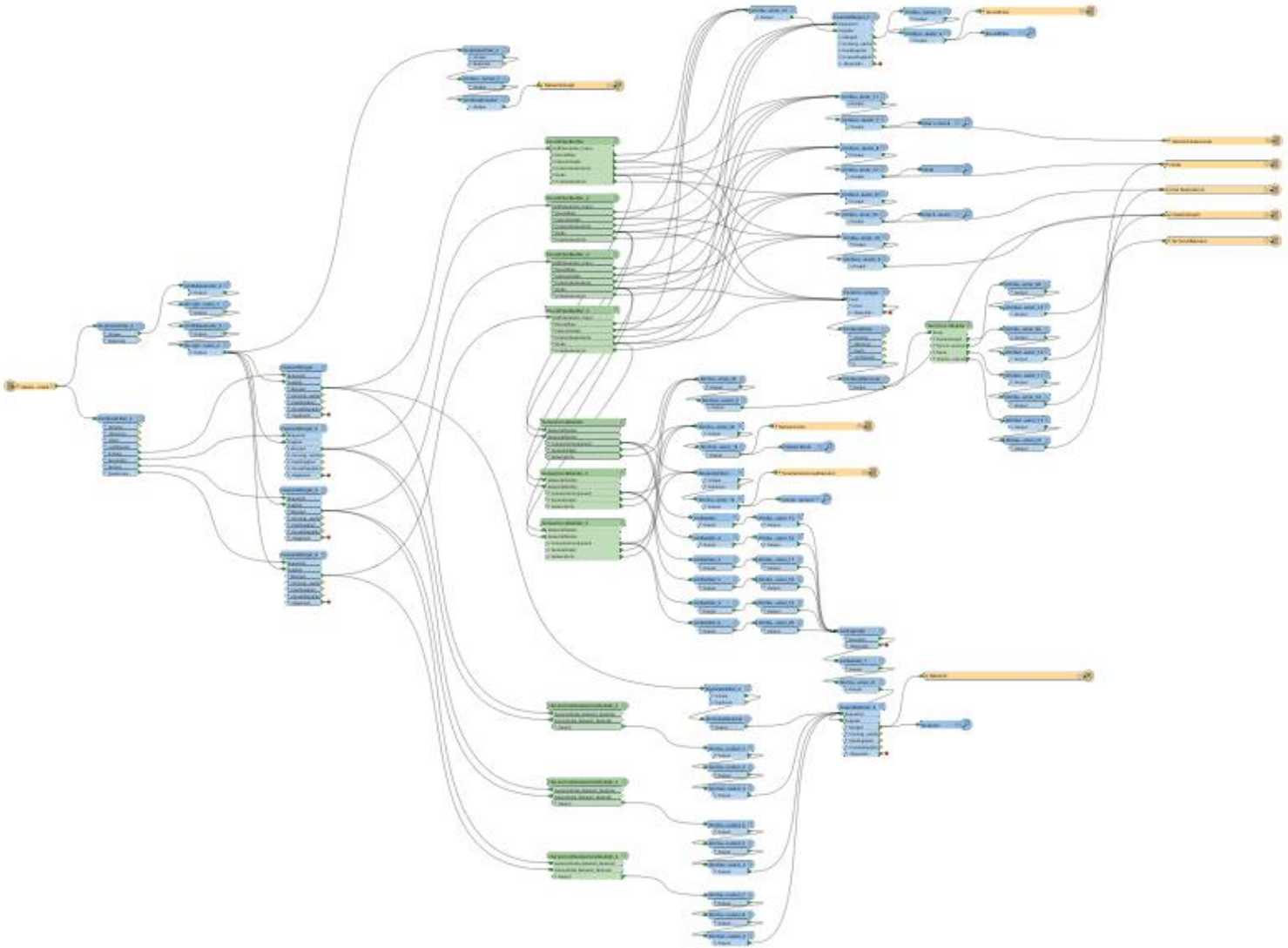
- To get people into using the UtilityNetwork ADE we have to start from the basics:
  - Making a small, basic network from a well-known source format
  - Performing basic routing analysis on said network
- We should highlight the UtilityNetwork ADE's strengths
  - Hierarchical network composition
  - Explicit topography and topology

Making a network

- FME workbench for creating hierarchical networks from ESRI shapefiles
  - Must not necessarily be in shapefile format, just a starting point
- FME workbench for importing this data into a 3DCityDB instance
- Current supports:

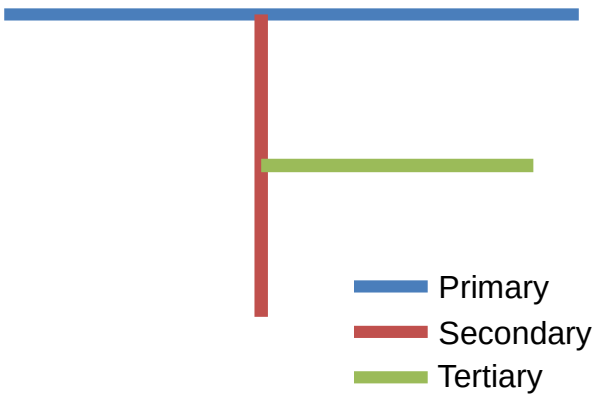
1. Network elements
2. NetworkGraph elements
3. RoundPipe Elements
4. Terminal Elements
5. FeatureGraph Elements
6. InteriorFeatureLink Elements
7. Node Elements
8. InterFeatureLink Elements
9. NetworkLink Elements

Current State



Hierarchical Networks

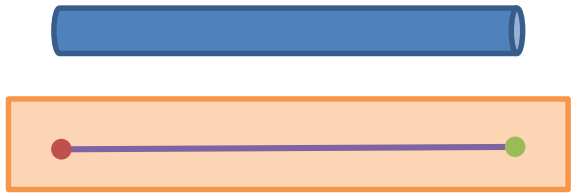
- Network elements are derived from attributes of RoundPipe elements
- NetworkGraph elements are stored as children of their respective Network elements
- Hierarchy level determined via an attribute, and are assigned as a child element of the Network element of higher order that it touches




```
<Network>
    <subOrdinateNetwork>
        <Network>
            <subOrdinateNetwork>
                <Network>
                </Network>
            </subOrdinateNetwork>
        </Network>
    </subOrdinateNetwork>
</Network>
```

RoundPipes

- RoundPipe elements are direct children of Network elements
- Their FeatureGraph elements are stored as children of the Network element's NetworkGraph element and referenced via xlink
- The FeatureGraphs contain two exterior Node elements and an InteriorFeatureLink element.

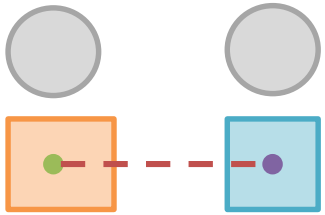


```
<Network>
    <topoGraph>
        <NetworkGraph>
            <featureGraphMember>
                <FeatureGraph>
                    <nodeMember>
                        <Node />
                    </nodeMember>
                    <nodeMember>
                        <Node />
                    </nodeMember>
                    <linkMember>
                        <InteriorFeatureLink>
                            <start xlink />
                            <end xlink />
                        </InteriorFeatureLink>
                    </linkMember>
                </FeatureGraph>
            </featureGraphMember>
        </NetworkGraph>
    </topoGraph>
    <component>
        <RoundPipe>
            <topoGraph xlink />
        </RoundPipe>
</Network>
```

EIFER

InterFeatureLinks

- InterFeatureLink elements are stored as children of their respective Network element's NetworkGraph element.
- They link Node elements found in network features' FeatureGraph Elements



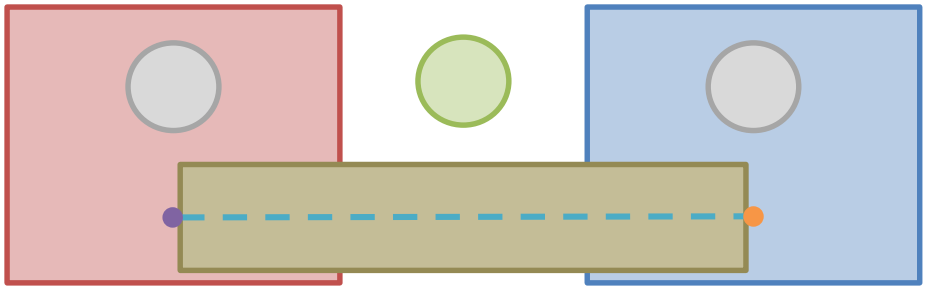
*Note: In reality, the network elements (grey circles) and Nodes (green and purple dots) are spatially coincident

```
<Network>
    <topoGraph>
        <NetworkGraph>
            <featureGraphMember>
                <FeatureGraph>
                    <nodeMember>
                        <Node>
                        </Node>
                    </nodeMember>
                    <nodeMember>
                        <Node>
                        </Node>
                    </nodeMember>
                </FeatureGraph>
            </featureGraphMember>
            <featureGraphMember>
                <FeatureGraph>
                    <nodeMember>
                        <Node>
                        </Node>
                    </nodeMember>
                </FeatureGraph>
            </featureGraphMember>
            <linkMember>
                <InterFeatureLink>
                    <start xlink />
                    <end xlink />
                </InterFeatureLink>
            </linkMember>
        </NetworkGraph>
    </topoGraph>
</Network>
```

EIFER

NetworkLinks

- NetworkLinks are links between separate networks
- They exist outside of any Network element, but link Node elements found within the FeatureGraph elements of elements in separate networks
- Should be modelled with a ConnectorElement, currently using SimpleFunctionalElement



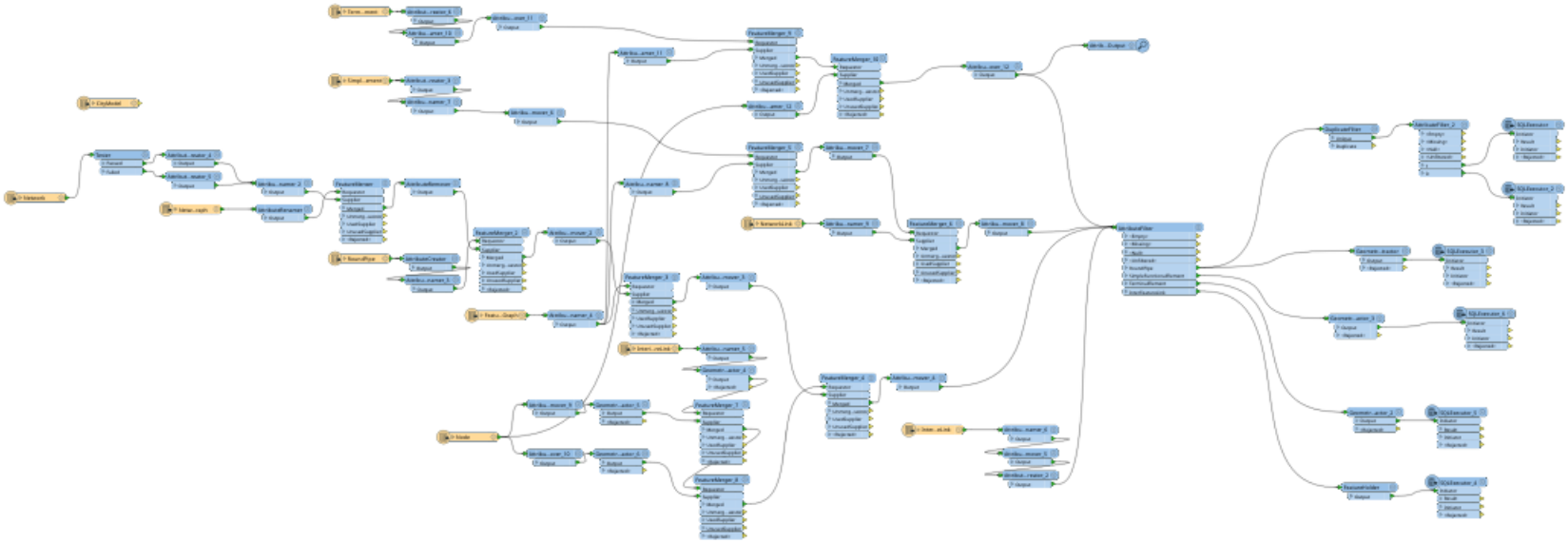
*Note: In reality, the network elements (grey & green circles) And Nodes (purple and orange dots) are spatially coincident

```
<core:cityObjectMember>
    <SimpleFunctionalElement>
        <topoGraph>
            <FeatureGraph>
                <networkLinkMember>
                    <NetworkLink>
                        <start xlink />
                        <end xlink />
                    </NetworkLink>
                </networkLinkMember>
            </FeatureGraph>
        </topoGraph>
    </SimpleFunctionalElement>
</core:cityObjectMember>
<Network>
    <topoGraph>
        <NetworkGraph>
            <FeatureGraph>
                <nodeMember>
                    <Node>
                    </Node>
                </nodeMember>
            </FeatureGraph>
        </NetworkGraph>
    </topoGraph>
</Network>
<Network>
    <topoGraph>
        <NetworkGraph>
            <FeatureGraph>
                <nodeMember>
                    <Node>
                    </Node>
                </nodeMember>
            </FeatureGraph>
        </NetworkGraph>
    </topoGraph>
</Network>
```

To the database and beyond

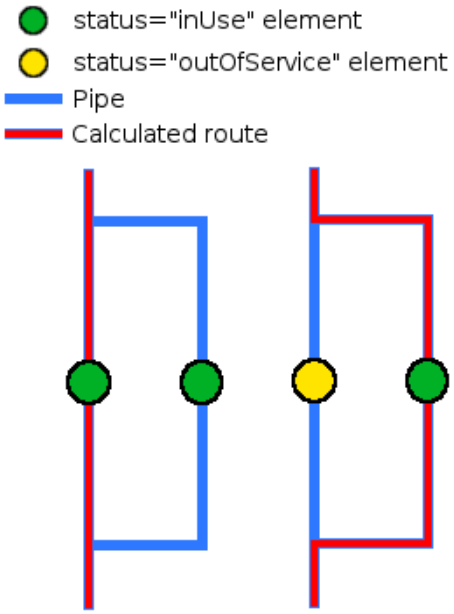
- FME workbench for taking the specific output of the first workbench and writing it into a 3DCityDB instance
  - Uses the database schema by Dr. Agugiaro, not the “auto-derived” schema from the importer/exporter

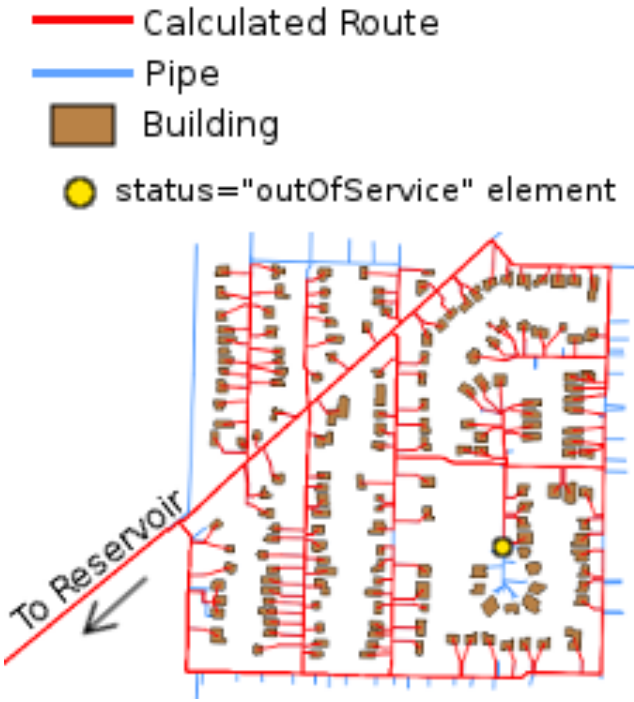
https://github.com/gioagu/3dcitydb_ade



pgRouting

- PostgreSQL extension for performing routing on topological graph structures
- Used in my master thesis





Python API For the UtilityNetwork ADE

- Started a Python API for interacting with the UtilityNetwork ADE
- (Almost) every pgRouting uses the same (or similar) “graph table” structure as an input
- The API creates a view out of the 3DCityDB (+UtilityNetwork ADE instance) which is used as the foundation for routing operations
- Currently somewhat basic, but implements two kinds of routing:
  - Dijkstra (one to one, one to many, many to one, many to many)
  - Flood fill (one, many)

https://github.com/iboates/UtilityNetwork-ADE-3DCityDB-Manager

Python API For the UtilityNetwork ADE

- Supports constriction of the network via exclusion of individual features or networks
- Can create “output tables” from routing analysis output for visualization (can also make them as views)
- With more work on getting/setting feature properties, this could become very powerful for functional modelling
- Stay tuned for a demo!

https://github.com/iboates/UtilityNetwork-ADE-3DCityDB-Manager

Known limitations

- Seems to have problems routing when a subnetwork has multiple connections to its parent network
- Does not support “lateral” connections (i.e. “sibling” networks with the same parent network)
- Does not support any supply attributes (fill level, flow rates, etc.)
- Only supports RoundPipe features for no

https://github.com/iboates/UtilityNetwork-ADE-3DCityDB-Manager

Going Forward

- My personal opinion is that the focus should be on making functional samples that can perform simple modelling tasks before defining exotic theoretical capabilities
- Also on documentation and best practices
- I do not personally have much time to devote to further development of this API, so I hope that there is interest among others to carry on this work

https://github.com/iboates/UtilityNetwork-ADE-3DCityDB-Manager

Thank you for your attention

M. Sc. Isaac Boates
Isaac.Boates@eifer.uni-karlsruhe.de

M. Sc. Alexandru Nichersu
Alexandru.Nichersu@eifer.uni-karlsruhe.de