Automated Generation of Simulation Models from Utility Network ADE Data

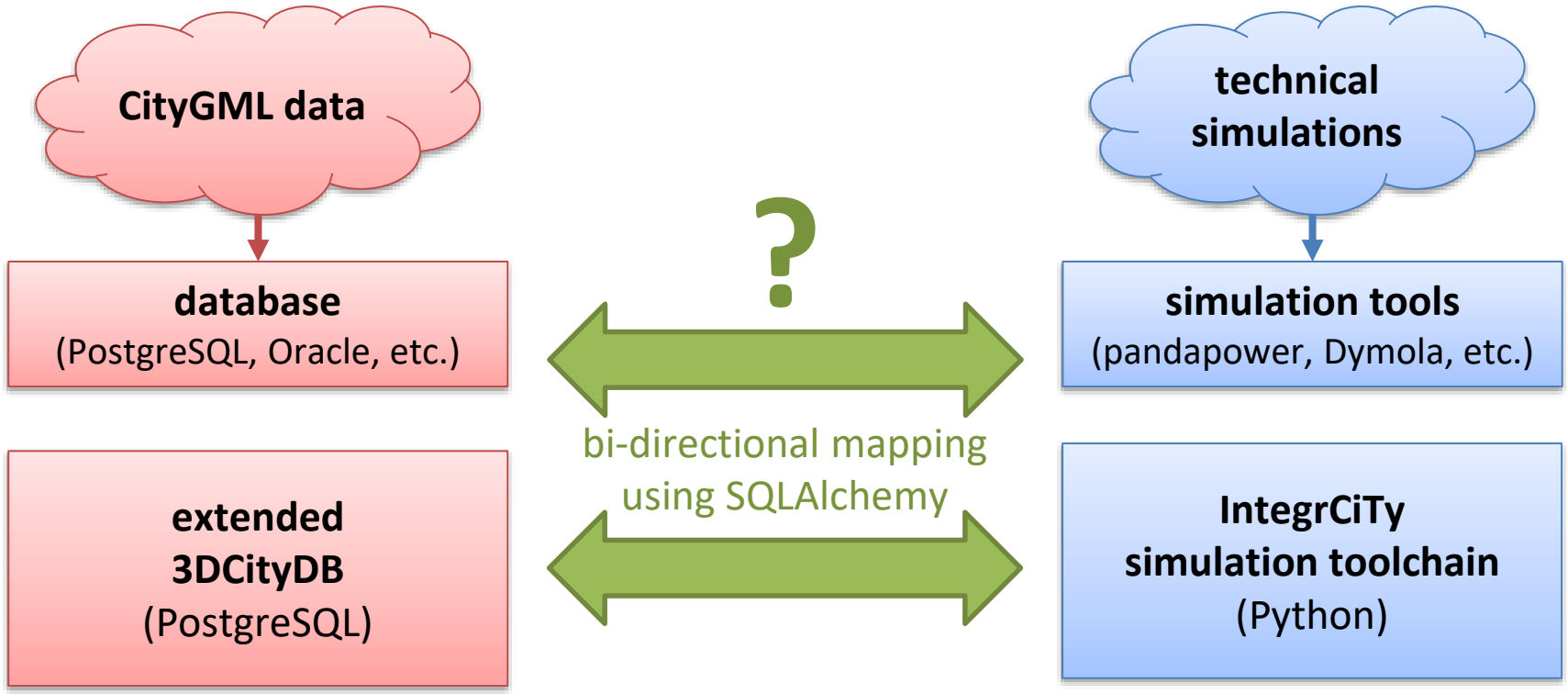
Edmund Widl[1], Giorgio Agugiaro[2], Pablo Puerto[3]

[1] AIT Austrian Institute of Technology, Center for Energy, Vienna, Austria

[2] Delft University of Technology, Department of Urbanism, Delft, The Netherlands

[3] CREM – Centre de Recherches Energetiques et Municipales, Martigny, Switzerland

Linking 3D semantic data models and simulation models



Package DBLayer – IntegrCiTy Data Access Layer

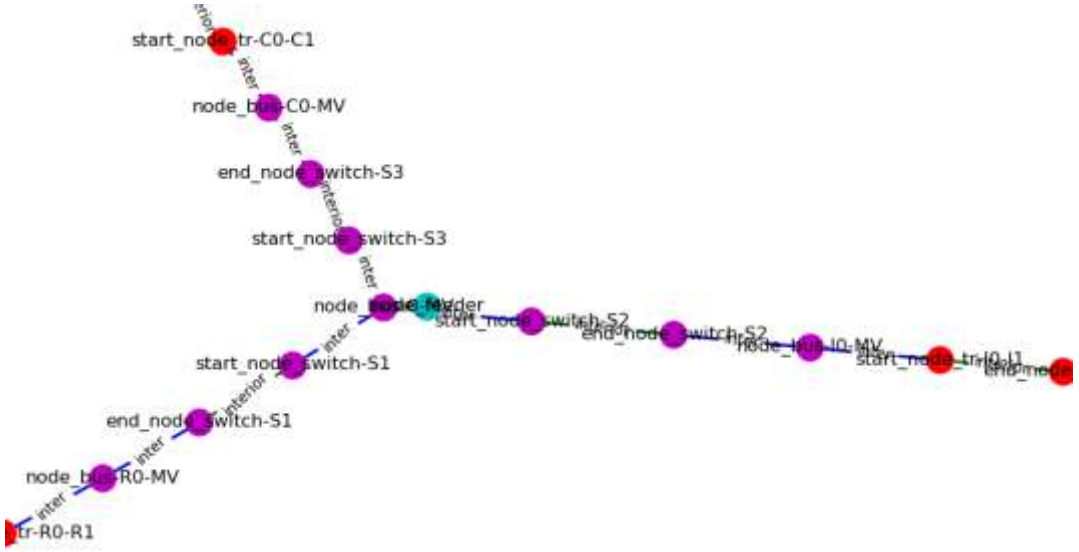
- intended purpose:
  - retrieving/storing scenario data from/to a database ➔ *extended 3DCityDB*
  - retrieving/storing co-simulation setups from/to a database ➔ *Simulation package* (not covered in this presentation)
  - creating simulation models from information stored in a database ➔ *tool-specific mappings*
- prototype implementation in Python
  - publicly available (open source): https://github.com/IntegrCiTy/dblayer

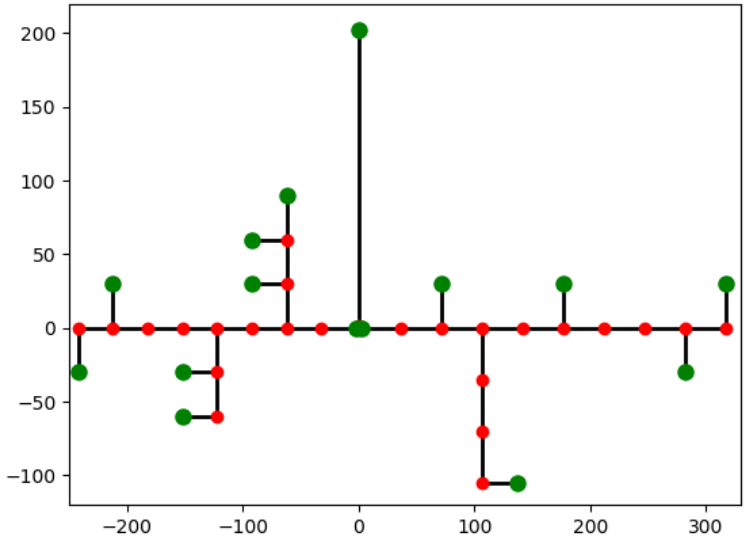
Package DBLayer – Examples (1/3)

- domain-specific functionality for storing power grid models
  - adapted to the thinking of power engineers:
    - start with electrical busses
    - connected busses with power lines
    - attach electrical loads to busses
    - etc.
  - applied to „CIGRE low-voltage test feeder“ (small but realistic distribution network model)
  - available online: https://github.com/IntegrCiTy/dblayer/blob/master/examples/PowerGridModelWriteDB.ipynb
- domain-specific functionality for storing thermal network models
  - similar to power grids, but not the same
  - available online: https://github.com/IntegrCiTy/dblayer/blob/master/examples/ThermalNetworkModelWriteDB.ipynb

Package DBLayer – Examples (2/3)

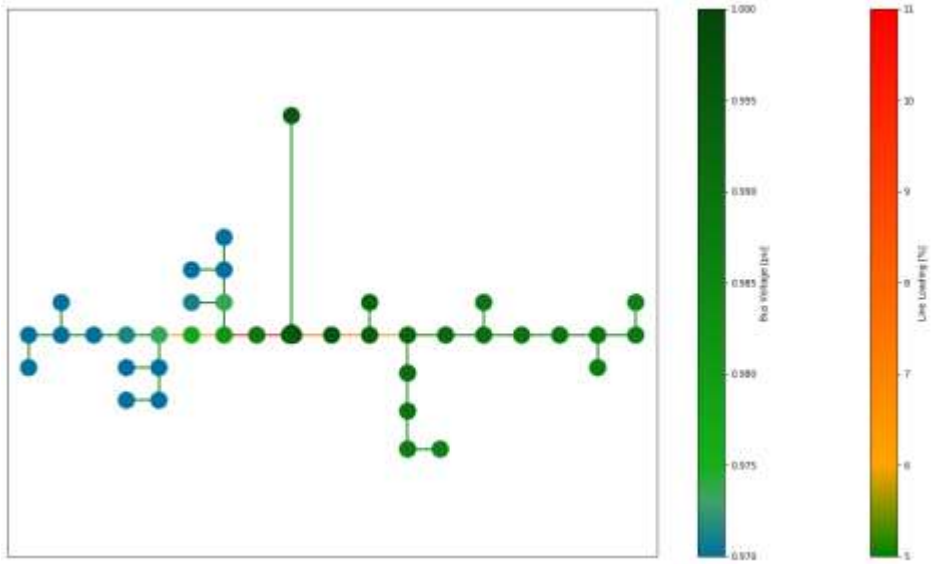
- retrieve and inspect data of a power grid model from the extended 3DCityDB
  - available online: https://github.com/IntegrCiTy/dblayer/blob/master/examples/PowerGridModelInspect.ipynb
- object-oriented approach built in top of package SQLAlchemy
  - automated mapping of data structures in the 3DCityDB to Python classes
  - retrieve data from standard tables (*citydb*) or views (*citydb_view*)
  - allows conditional queries and joining of data over two or more tables / views

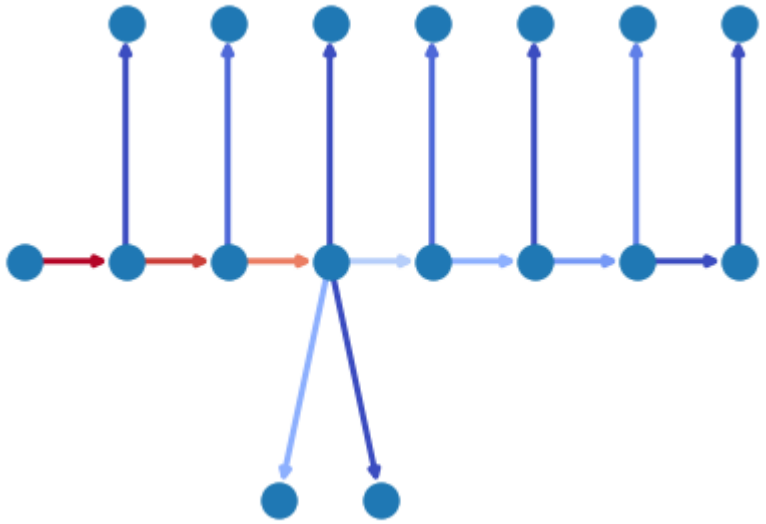




Package DBLayer – Examples (3/3)

- automatedly create a simulation model from a power grid model from the extended 3DCityDB
  - available online: https://github.com/IntegrCiTy/dblayer/blob/master/examples/PowerGridModelSim.ipynb
- automatedly create a simulation model from a thermal network model from the extended 3DCityDB
  - available online: https://github.com/IntegrCiTy/dblayer/blob/master/examples/ThermalNetworkModelSim.ipynb





General findings and learnings

- UN ADE is definitely capable of representing different types of networks in a way that allows to extract simulation models
- functions and views provided by Giorgio‘s extended 3DCityDB are a valuable tool for implementing the required functionality
- some practical challenges remain:
  - some attributed and classes are missing

    ➔ see example for power grids on the next slides
  - mapping of available (or de facto) standards to UN ADE would definitely be a prerequisite for the adoption by domain experts

    ➔ e.g., mapping of CIM to UN ADE classes / attributes

Suggestions for improvements regarding power grid models

- two key concepts for representing power grids are not very well covered (or probably I missed something in the UML diagrams)
  - **electrical busses**
    - fundamental building blocks of power grids
    - mapping to *SimpleFunctionalElement* is an option, but lacks some details:
      - *type*: node, busbar, muff, …
      - *grid voltage level* (+unit): if not as attribute, *ElectricalMedium* associated to *Network* might work, but would require to split a power system in several networks
  - **electrical loads**
    - I used *TerminalElement* associated to *ElectricalAppliances* (Energy ADE), but this lacks an important detail:
      - *reactive power* (+unit)
      - probably somehow using *ElectricalMediumSupply* (negative supply)?

- **lines & cables**
  - missing attribues:
    - *resistance* (+unit)
    - *reactance* (+unit)
    - *capacitance* (+unit)
    - *maximum thermal loading* (+unit)
  - possible alternative: new class representing a „line type“ (avoids unnecessary replication of data)
- **transformers**
  - I used class *ComplexFunctionalDevice*, but …
  - needs special treatment, because of required semantic differentiation between low-voltage and high-voltage side
  - possible alternative: combination of device and network link
    - compare with situation of grid voltage level for electrical busses

Conclusion and outlook

- Initial work to bridge the gap between “GIS” and “technical-simulations” worlds
- acquired reasonable understanding of **requirements for power grids**
  - mapping to more simulators needed (at least conceptually) for better understanding
- only rough idea for **thermal networks** and **gas networks**
  - lack of generally agreed upon “test systems”
  - probably not an obstacle but rather an opportunity?
- currently implemented in project **IntegrCiTy**, but…
- …developed tools are **generic and flexible** enough to be used also in other contexts
  - *intentional* open development → test and give (constructive) feedback

THANK YOU!

Edmund Widl