Shimmer software architecture proposal

Matteo Cicuttin

DISMA - PoliTO

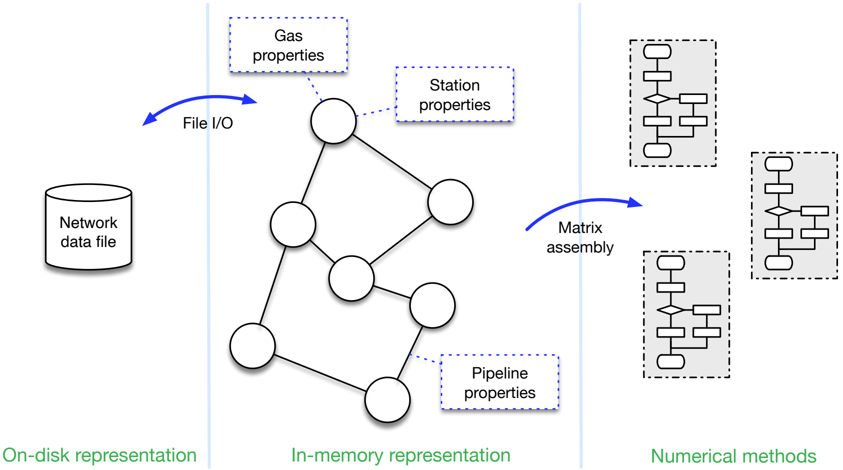
September 18, 2023

Context

Goal: Build a tool to simulate multi-gas distribution networks

- Open source tool for public dissemination
- Open & documented data exchange format
- Interchangeable numerical models

System boundaries



File I/O

Gas properties

Station properties

Pipeline properties

Network data file

Matrix assembly

On-disk representation

In-memory representation

Numerical methods

Data exchange and on-disk representation

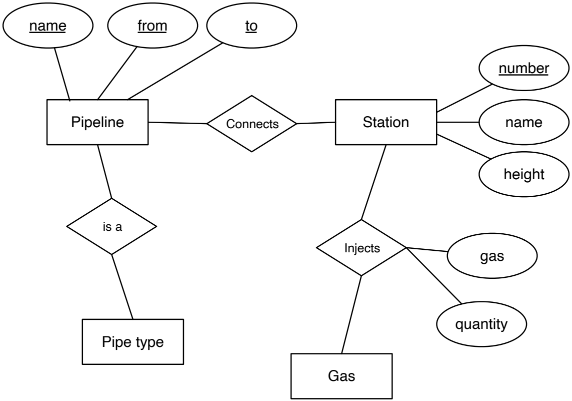
Exchanging data is frequently a challenge. We want to get this right from the beginning:

- Gas network data should be stored in an **open & well supported format**
- Data format should guarantee **data correctness** and **integrity**. For example:
  - impossible to insert a pipe between non-existent stations
  - impossible to remove a station with attached pipes
  - impossible to inject a gas from a non-existent station

SQLite fullfills all the requirements:

- Widely supported on all main OSs and by most of the scientific tools. Example: native support in Matlab, plug-ins for Octave and R
- Full fledged SQL database, data constraints easily specified & enforced
- Graphical tools for data manipulation & import/export exist

Oversimplified relational data model



Relational data models are ubiquitous and well-understood:

- Clear and unambiguous **entities** and **relations**
- Data integrity automatically checked: impossible to enter an edge if a node does not exist
- We need to discuss the actual data model offline in order to determine if it fits all the requirements

Stations

```sql
-- The stations. They are the nodes of the graph
create table stations (
    s_name      TEXT,
    s_number    INTEGER,
    s_height    REAL,
    PRIMARY KEY(s_number)
);
```

Pipelines

```
-- The pipelines. They are the edges of the graph.
create table pipelines (
    p_name      TEXT NOT NULL,
    s_from      INTEGER,
    s_to        INTEGER,
    p_type      INTEGER,
    PRIMARY KEY (p_name, s_from, s_to),

    -- The source station must exist
    FOREIGN KEY (s_from)
        REFERENCES stations(s_number),
    -- The destination station must exist
    FOREIGN KEY (s_to)
        REFERENCES stations(s_number)
    -- The pipeline type must be valid
    FOREIGN KEY (p_type)
        REFERENCES pipeline_types(p_type)
);

```

Pipeline element types

```sql
-- Pipeline element type. Can be a pipe, a compressor, a regulator, ...
create table pipeline_types (
    p_type      INTEGER,
    t_name      TEXT NOT NULL,
    PRIMARY KEY (p_type)
);
insert into pipeline_types values (0, 'pipeline');
insert into pipeline_types values (1, 'resistor');
insert into pipeline_types values (2, 'compressor');
insert into pipeline_types values (3, 'regulator');
insert into pipeline_types values (4, 'valve');
```

Pipeline parameters (element type 0)

```sql
-- Pipeline parameters as length, diameter and so on.
create table pipeline_parameters (
    p_name      TEXT_NOT_NULL,
    s_from      INTEGER,
    s_to        INTEGER,
    length      REAL NOT NULL,
    diameter    REAL NOT NULL,
    epsi        REAL NOT NULL,

    -- The referenced pipeline must exist
    FOREIGN KEY (p_name, s_from, s_to)
        REFERENCES pipelines(p_name, s_from, s_to)
);
```

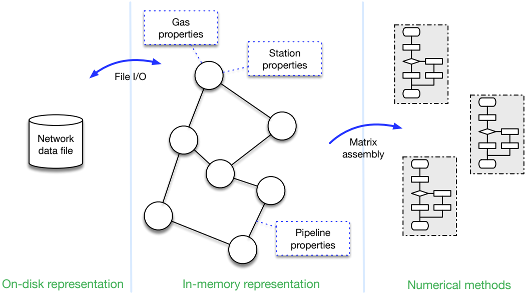
Who injects what

```
-- Who injects what
create table injects (
    s_number    INTEGER,
    g_name      TEXT,
    quantity    REAL,

    -- station number must be valid
    FOREIGN KEY (s_number)
        REFERENCES stations(s_number),
    -- gas name must be valid
    FOREIGN KEY (g_name)
        REFERENCES gases(g_name)
);

-- The gases. Which are the parameters associated to each gas?
create table gases (
    g_name      TEXT,
    PRIMARY KEY(g_name)
);

```

In-memory representation

The in-memory representation is a labelled graph.

- Decouples away data handling both for input and for output
- Matrices for numerical methods are directly built from graph
- Easily implemented with `boost::graph`, graph algorithms for free



Technologies

We chose industry standard, portable and widespread technologies.

- SQLite for data storage and exchange: widespread format, extremely portable
- Data manipulation and processing
  - `boost::graph` for graph manipulation
  - Eigen for linear algebra and numerical methods
- Qt to have a portable graphical toolkit and easy to install development environment

Conclusions

Next tasks:

- Iterate on the data model and finalize its design (DENERG/DISMA)
- Provide formatted & cleaned-up data in Excel/CSV files (DENERG)
- Provide the first two layers of the system (DISMA)
- Re-implement Matlab stuff in the new architecture (DENERG/DISMA)
- Validate the implementation