

# NetworkSimDB – Simulation database for networks with associated visualization

Matthias König, Michael Weidlich & Torsten Houwaart

## Introduction

**NetworkSimDB** is a database for (network) graphs and their associated node and edge information.

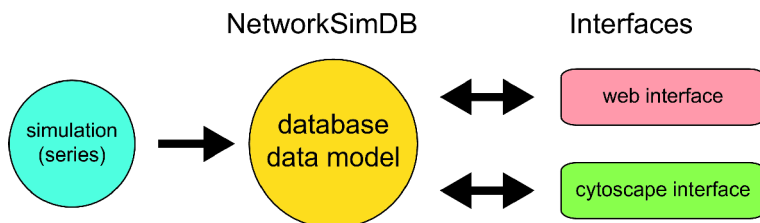
**NetworkSimDB** comprises of three main parts: 1. database, 2. web interface and 3. additional visualization capabilities for the graphs and the information.

Possible applications of **NetworkSimDB** are the storage and analysis of simulation results based on flux methods like FBA or kinetic modelling in metabolic networks. The system is not limited to metabolic networks but is a flexible tool for the work with graphs with associated information like they occur in a wide range of applications (social networks, protein-interaction networks, ...).

The information can be stored in the database via API access or via web interface or the Cytoscape Interface. The communication with the database is controlled via a Hibernate ORM layer.

**NetworkSimDB** is implemented in Java and available under GNU public license (GPL) at <http://sourceforge.net/projects/networksimdb/>

This project's SourceForge.net Subversion repository can be checked out through SVN `svn co https://networksimdb.svn.sourceforge.net/svnroot/networksimdb networksimdb`



**Fig.1 NetworkSimDB.** Overview information flow between data generation (graphs with associated information) the database (persistent storage) and the interfaces for the visualization of the database content. A web interface and a Cytoscape Plugin for data retrieval and data storage are developed.

## Data Model

Core tables of the data model are the Graph, Node, Edge, Mapping and Collection (overview of data model in Fig.2). Due to a very general graph model, the data model can be used for the representation of nearly any type of graph. Unlimited associated information in form of graph information, node information or edge information can be stored.

**Graph:** A Graph is represented as a set of nodes and edges. The Graph data model can be applied to directed and undirected graphs and even multigraphs.

**Edge:** Edges are directed edges from source Node to target Node (source → target). Undirected edges (source ↔ target) are represented by the the edge tuple (source → target) and (target → source).

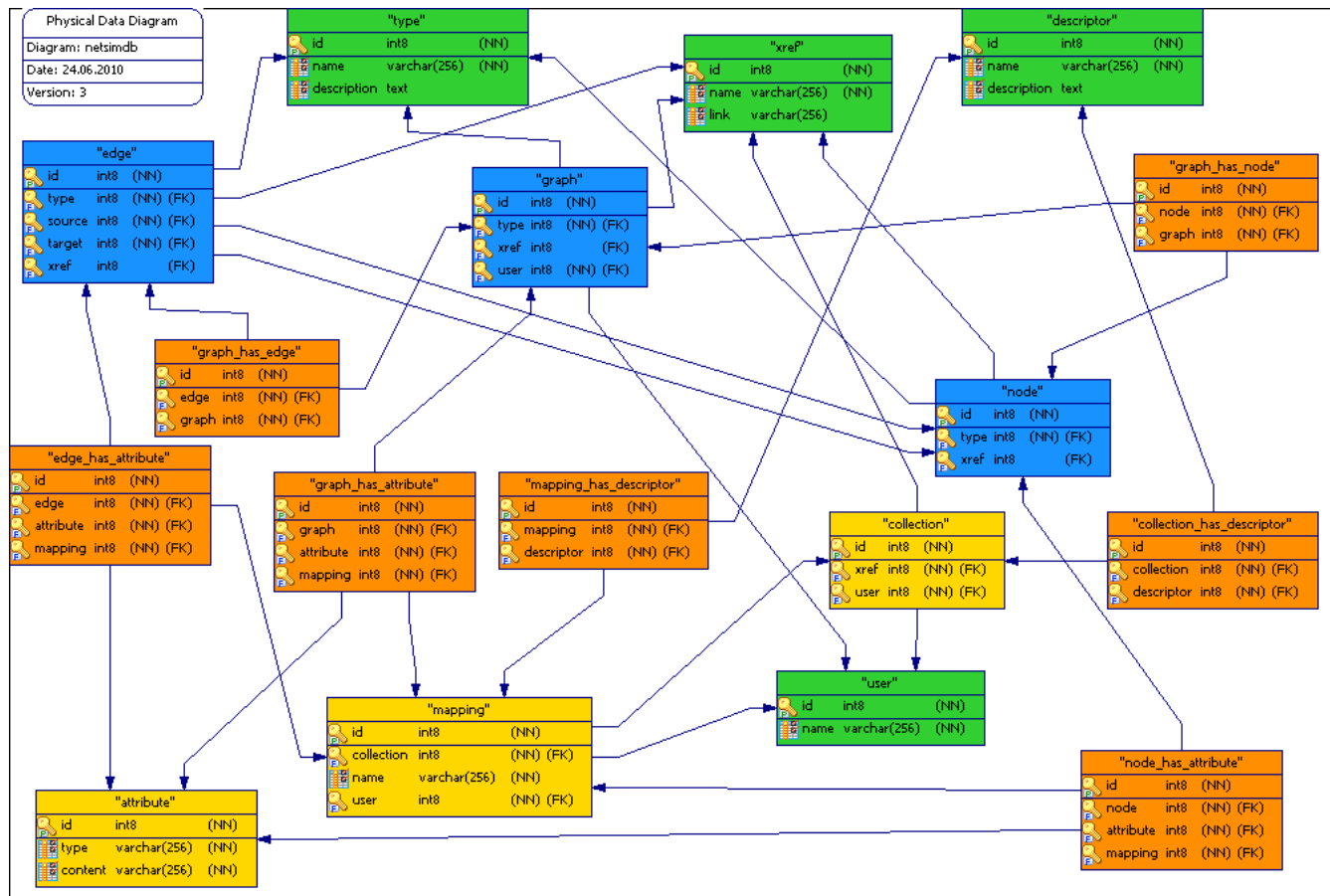
**Attributes:** An attribute is an stored bit of information. Attributes can be associated to nodes, edges or graphs. The type of attribute (Integer, Double, String, Boolean) defines which type of information is stored in the attribute.

**Mapping:** A mapping is a set of attributes which are associated to nodes or edges. Depending of the type of the mapping NodeMappings and EdgeMappings can be discriminated. A mapping maps one type of information on the graph (nodes or edges). For example the information 'interaction strength' on the edges of the network.

**Collection:** A collection is set of Mappings (Node- and EdgeMappings). A Collection can be seen as all

information necessary to generate a full image of the graph.

In addition to the core classes tables for the storage of user information (**User**) and additional information (**Xref**, **Description**) are implemented.



**Fig.2 Data model.** Overview data base model and tables. Tables related to the Graph are shown in blue (graph, node, edge). Tables for the storage of associated graph information are shown in yellow (attribute, mapping, collection). Tables for additional information are green (type, xref, descriptor, user). N x M tables in red.

## Data Base

**NetworkSimDB** is independent from the underlying database layer due to the use of JDBC drivers. In principle the data base model can be implemented on any database system supporting JDBC for Java.

**NetworkSimDB** has been tested with PostgreSQL and SQLite.

## Hibernate Mapping

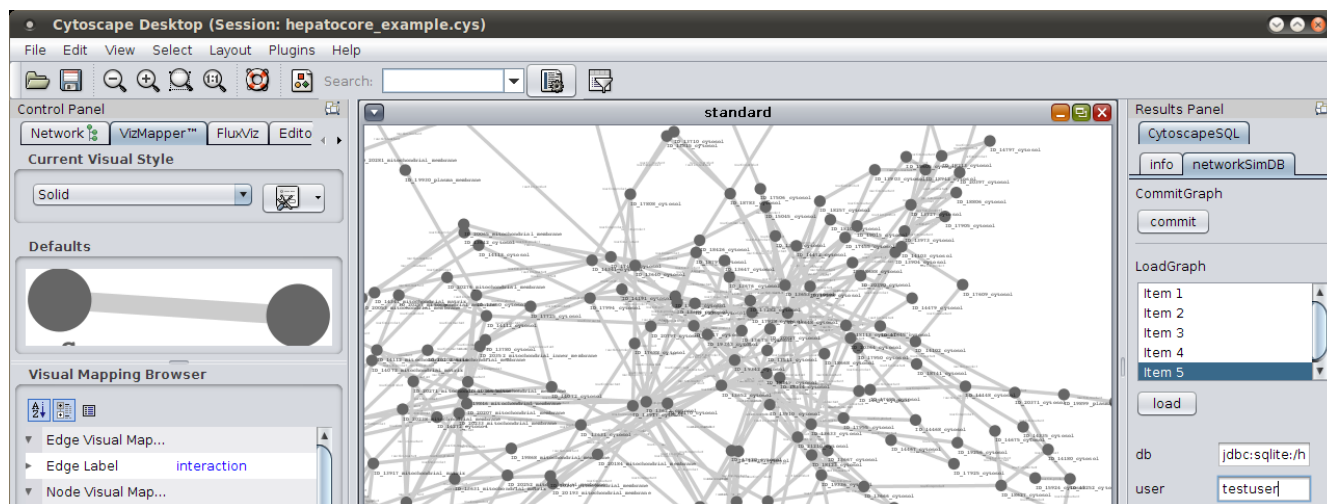
The main database tables are mapped to plain old java objects via an intermediate Hibernate layer. The Hibernate layer is used for programmatic access to the database content.

## Web Interface

**NetworkSimDB** is accessible via a web interface. Networks and Attributes in a variety of formats can be loaded in the database. The Collections, Mappings and Graphs can be visualized using this web interface.

## Cytoscape Interface

In addition to the web interface the database content can be visualized using Cytoscape. For this purpose an Cytoscape Plug-In has been developed which implements the communication between the visualization layer and the database. Like in the web interface the Hibernate mapping is used for data persistence and integrity.



**Fig.3 Cytoscape interface.** Graphs and graph information can be loaded in Cytoscape in a variety of formats (SBML, GraphML, SIF, PSI-PI). The loaded graphs can be stored in **networkSimDB** via the networkSimDB Panel on the right. Graphs and associated information can be loaded from networkSimDB and be visualized in Cytoscape using the VisualMapper on the left.