A geospatial relational database schema for interdependent network analysis and modelling

David Alderson[*1], Stuart Barr[†1], Tomas Holderness[‡2],
Craig Robson[^1] and Alistair Ford[#1]

Corresponding author: David Alderson (david.alderson@ncl.ac.uk)

[1]School of Civil Engineering and Geosciences
Cassie Building, Newcastle University, NE1 7RU, UK

[2]SMART Infrastructure Facility
Faculty of Engineering and Information Sciences
University of Wollongong, NSW, 2522, Australia

Summary

Services delivered via National Infrastructure (NI) are key to securing economic growth and societal well-being. Spatially complex interdependent networks form an integral component of NI (e.g. energy supply, transport, waste management, clean water supply and dirty water treatment). It is essential that such infrastructure networks and interdependencies can be managed, analysed and modelled in a robust and consistent manner. This paper presents work undertaken to develop a spatial interdependent network model within existing relational database management software that is suitable for national-scale representation of infrastructure network systems and their interdependencies.

KEYWORDS: National infrastructure, networks, relational databases, data management, interdependency.

1. Introduction

National Infrastructure (NI) networks no longer operate as isolated, stand-alone spatial networks, but interact to form complex relationships between physical assets and networks. Spatial dependencies and interdependencies between NI networks can lead to the propagation of disruptions and failures resulting from man-made and natural hazards. As such, it is pertainet to consider NI as as a 'network-of-networks', with inherent underlying spatiality. However, such a representation can require large quantities of data from disparate sources to represent physical components of single and multiple networks. Moreover, new data management and analysis tools are required in order to effectively manage and analyse infrastructure systems within the network-of-networks paradigm. In this paper we present a new relational database schema, implemented in PostgreSQL and PostGIS, that has been developed specfically for representing spatially interdependent infrastructure networks. The utility of the database schema is demonstrated via a prototype reporting tool for infrastructure network visualisation.

[*] David.Alderson@ncl.ac.uk
[†] Stuart.Barr@ncl.ac.uk
[‡] Tomas@uow.edu.au
[^] C.A.Robson1@ncl.ac.uk
[#] Alistair.Ford@ncl.ac.uk

2. Network database schema and Python interface

The structure of the network database schema is illustrated in **Figure 1** as an entity-relationship (ER) diagram. The administrative tables, *Graphs* and *Global_Interdependency* (**Figure 1**) maintain a record of the networks and interdependencies stored within the schema. A network is stored using the concept of table inheritance. A single network comprises three tables, with each inheriting attributes from the *parent* tables (**Figure 1**); ensuring that the attributes of the *parent* tables propagate to the *instance* tables. This same approach is used for the interdependency links between two networks.

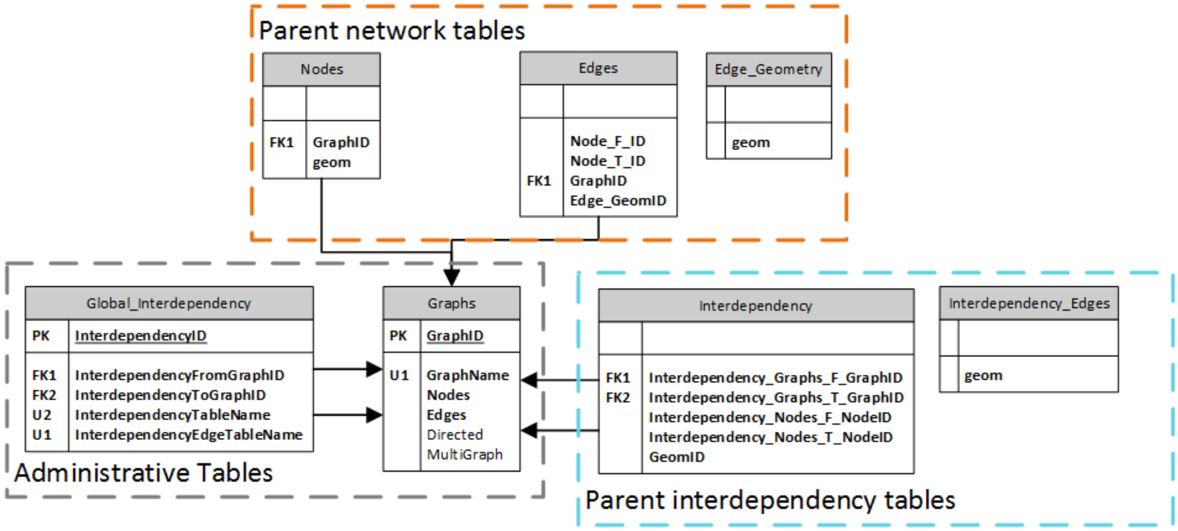

Parent network tables
Nodes
FK1 GraphID
geom
Edges
Node_F_ID
Node_T_ID
FK1 GraphID
Edge_GeomID
Edge_Geometry
geom
Global_Interdependency
PK InterdependencyID
FK1 InterdependencyFromGraphID
FK2 InterdependencyToGraphID
U2 InterdependencyTableName
U1 InterdependencyEdgeTableName
Administrative Tables
Graphs
PK GraphID
U1 GraphName
Nodes
Edges
Directed
MultiGraph
Interdependency
FK1 Interdependency_Graphs_F_GraphID
FK2 Interdependency_Graphs_T_GraphID
Interdependency_Nodes_F_NodeID
Interdependency_Nodes_T_NodeID
GeomID
Interdependency_Edges
geom
Parent interdependency tables

Figure 1 – ER diagram of the network database schema.

Figure 2 shows two networks stored within the network database schema; the tables outlined in magenta represent a network derived from National Grid electricity network data while the tables outlined in green represent a network forming the London Underground Tube Network.

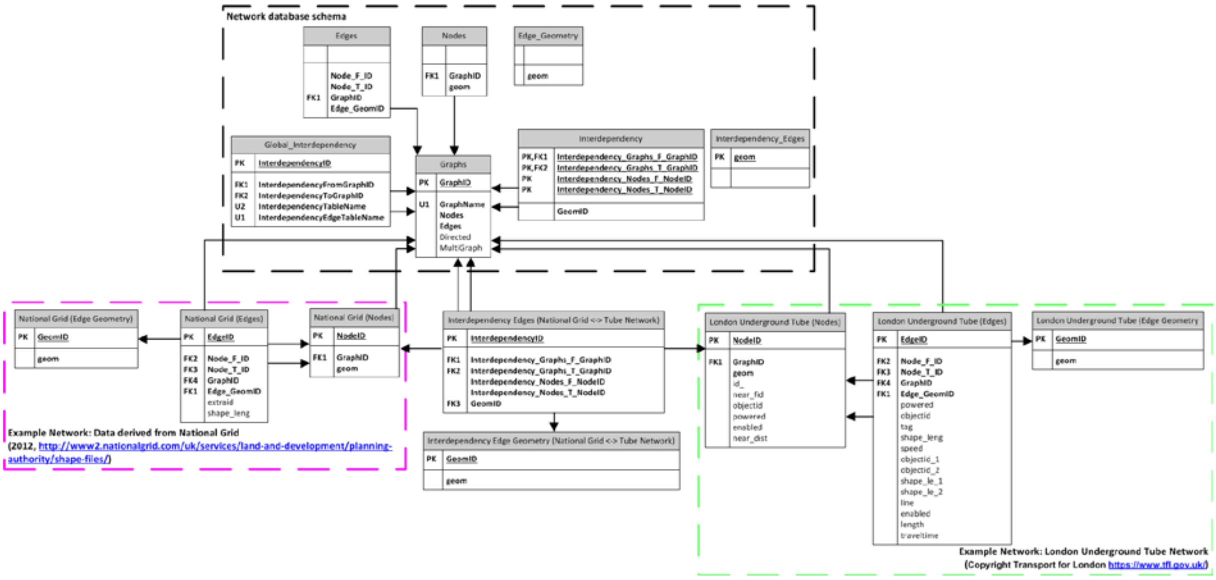

Network database schema
Edges
Node_F_ID
Node_T_ID
FK1 GraphID
Edge_GeomID
Nodes
FK1 GraphID
geom
Edge_Geometry
geom
Global_Interdependency
PK InterdependencyID
FK1 InterdependencyFromGraphID
FK2 InterdependencyToGraphID
U2 InterdependencyTableName
U1 InterdependencyEdgeTableName
Graphs
PK GraphID
U1 GraphName
Nodes
Edges
Directed
MultiGraph
Interdependency
PK,FK1 Interdependency_Graphs_F_GraphID
PK,FK2 Interdependency_Graphs_T_GraphID
PK Interdependency_Nodes_F_NodeID
PK Interdependency_Nodes_T_NodeID
GeomID
Interdependency_Edges
PK geom
National Grid (Edge Geometry)
PK GeomID
geom
National Grid (Edges)
PK EdgeID
FK2 Node_F_ID
FK3 Node_T_ID
FK4 GraphID
FK1 Edge_GeomID
extraid
shape_leng
National Grid (Nodes)
PK NodeID
FK1 GraphID
geom
Example Network: Data derived from National Grid
(2012, http://www2.nationalgrid.com/uk/services/land-and-development/planning-authority/shape-files/)
Interdependency Edges (National Grid <-> Tube Network)
PK InterdependencyID
FK1 Interdependency_Graphs_F_GraphID
FK2 Interdependency_Graphs_T_GraphID
Interdependency_Nodes_F_NodeID
Interdependency_Nodes_T_NodeID
FK3 GeomID
Interdependency Edge Geometry (National Grid <-> Tube Network)
PK GeomID
geom
London Underground Tube (Nodes)
PK NodeID
FK1 GraphID
geom
id_
near_fid
objectid
powered
enabled
near_dist
London Underground Tube (Edges)
PK EdgeID
FK2 Node_F_ID
FK3 Node_T_ID
FK4 GraphID
FK1 Edge_GeomID
powered
objectid
tag
shape_leng
speed
objectid_1
objectid_2
shape_le_1
shape_le_2
line
enabled
length
traveltime
London Underground Tube (Edge Geometry
PK GeomID
geom
Example Network: London Underground Tube Network
(Copyright Transport for London https://www.tfl.gov.uk/)

Figure 2 –ER diagram of two dependent networks stored in the database schema.

To construct and store the two networks, two Python-based modules, *nx_pg* and *nx_pgnet* have been developed that use NetworkX to build the topology of any network model (**Figure 3**). The *nx_pg* module is used to convert the raw data into a NetworkX network using the *read_pg()* function, and then the *nx_pgnet* module can be used to write this network to a set of schema-enabled tables back to the same database, using the *write.pg_net()* function. The process of building from raw data is illustrated in **Figure 3** by the steps highlighted in red, whilst the process of extracting a network from the schema, once built, is highlighted in green. The build process uses the *add_node()* and *add_edge()* methods of NetworkX to add the geometry and attributes of the raw edge and node data to a NetworkX network before the final result is then written to the database schema. These modules therefore act as a linking mechanism between the schema used for storage, and NetworkX which is used for network analysis.

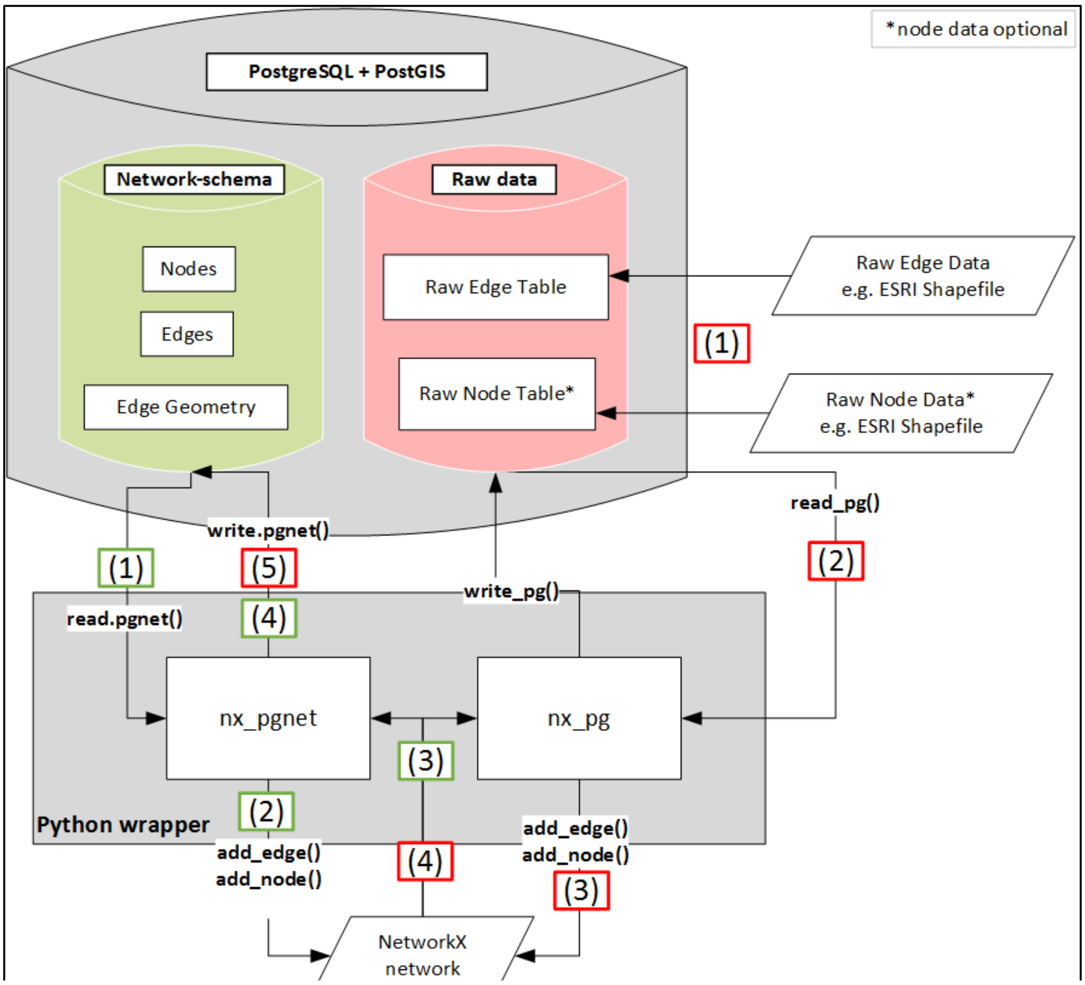


Figure 3 – Overview of Python modules *nx_pg* and *nx_pgnet* that act as an interface between the network database schema and NetworkX.

3. Enabling network analysis and visualisation

To date, the database schema has been used to facilitate the modelling and subsequent analysis of infrastructure networks at differing spatial scales, for different geographic regions, within different specific application domains. For example, the construction and storage of a road network from informal data sources within the database schema, and subsequent linkage to NetworkX via the Python wrappers, formed a critical component of work assessing optimum faecal sludge removal and disposal over a road network for a suburb in Kibera, Nairobi, Kenya (Kennedy-Walker et al., 2014). This work focussed on assessing two approaches for transferring faecal sludge between public toilets and transfer stations; namely via a single transfer station, or via multiple transfer stations (see (a) and (b) of **Figure**

4), using the network database schema as the data storage and analysis platform linked to NetworkX. This provided the facility to perform cost analyses based around shortest paths across the road network for the two different transfer approaches, between the toilets and transfer stations. In addition to its analytical utility, two database-level functions, *ni_create_node_view*, and *ni_create_edge_view*, create *views* of the network data allowing the schema to be directly linked to Quantum GIS (Quantum GIS, 2014). This functionality is shown in **Figure 5** where the two networks in Figure 2 (National Grid electricity transmission and London underground tube network) are shown along with a representation of inferred spatial dependencies between these two networks.

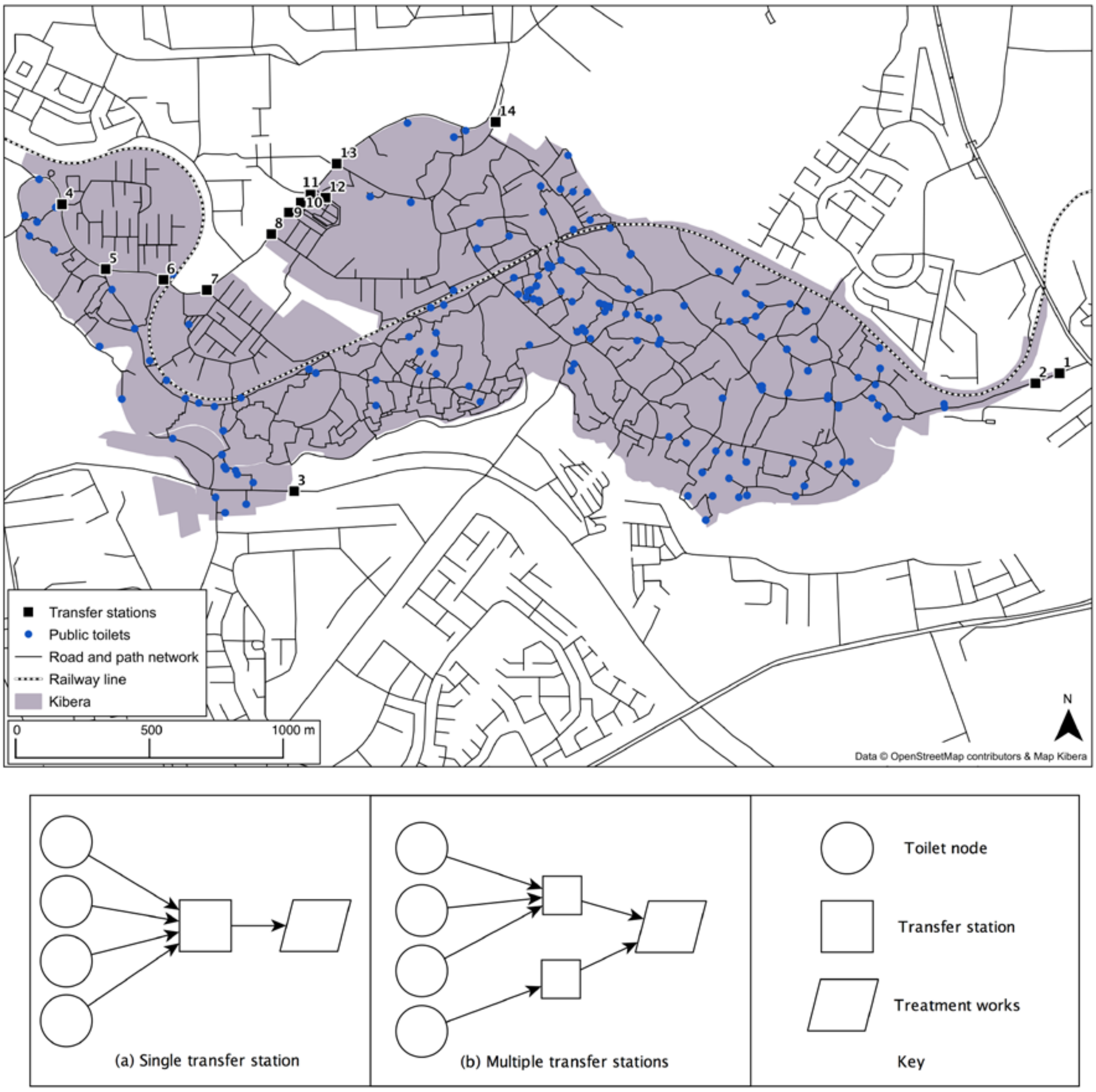
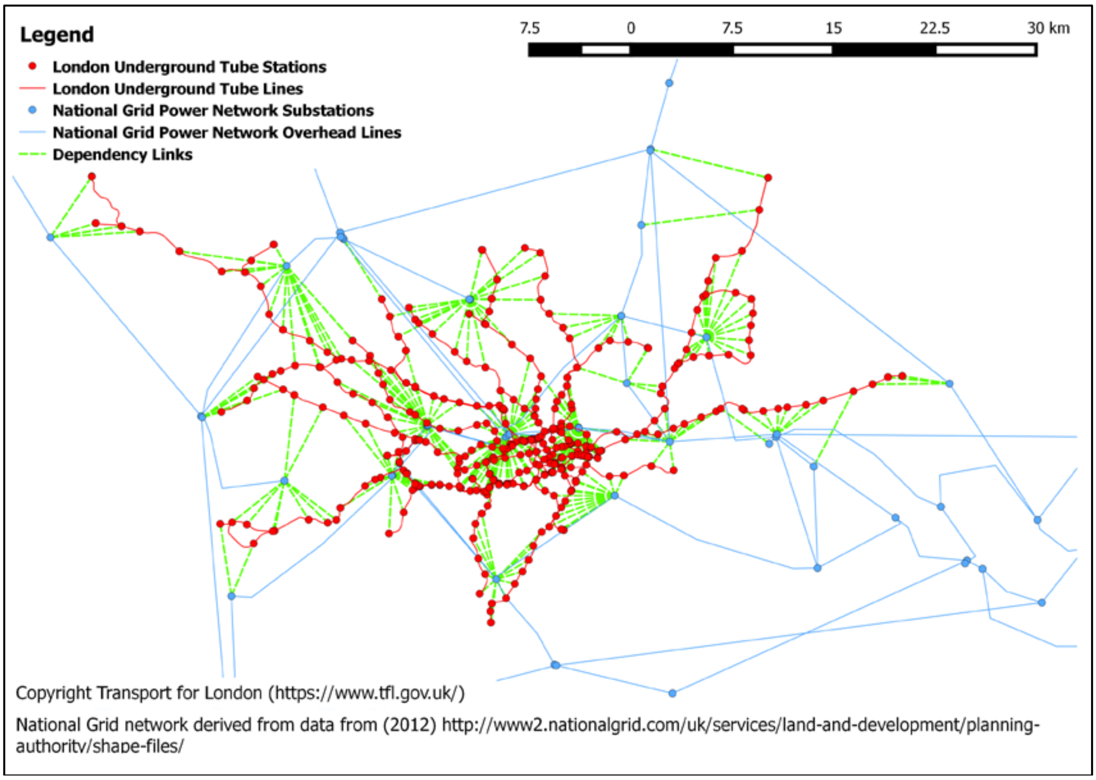


Figure 4 – Road network for Kibera suburb, Nairobi, Kenya (Kennedy-Walker et al., 2014).

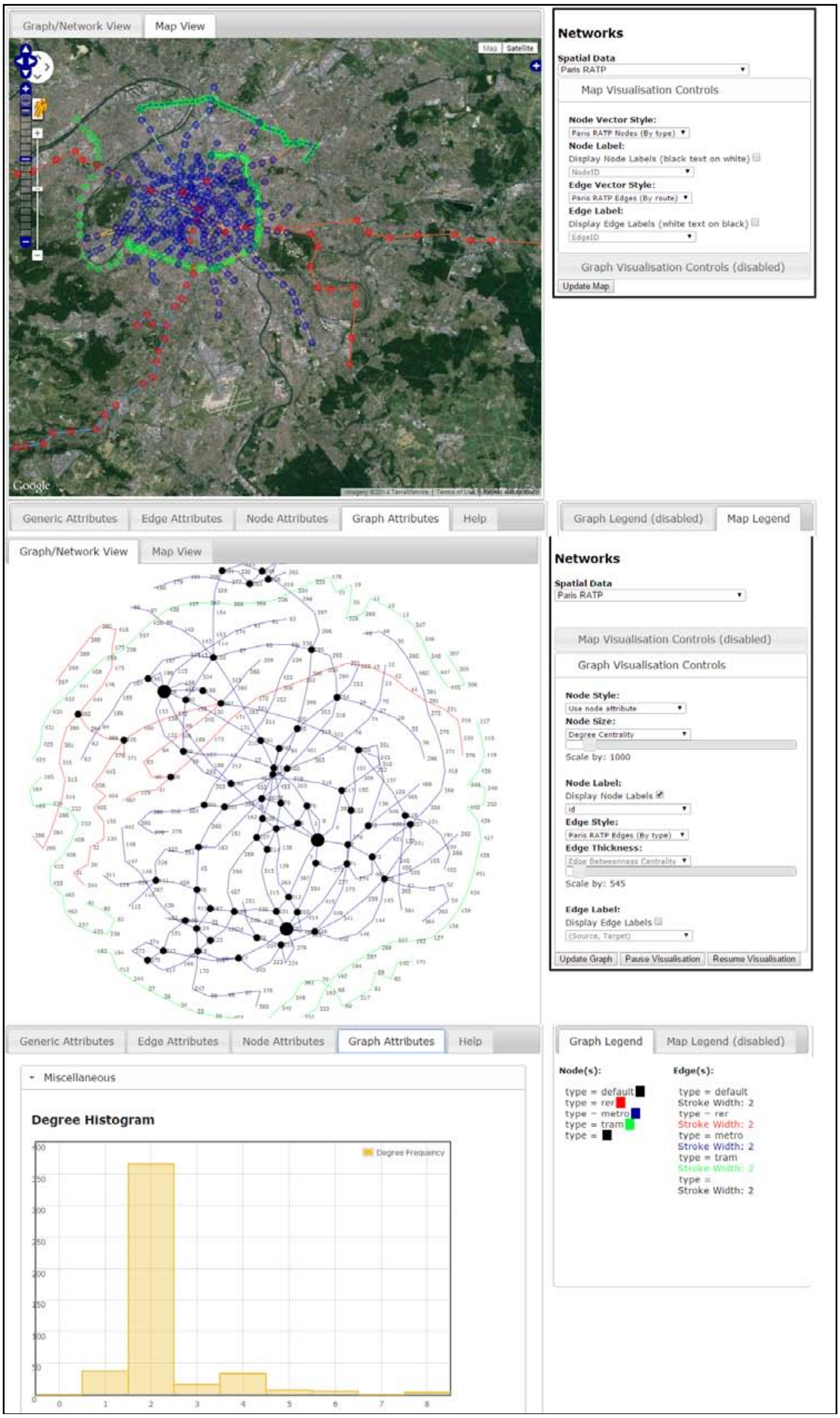

Legend
London Underground Tube Stations
London Underground Tube Lines
National Grid Power Network Substations
National Grid Power Network Overhead Lines
Dependency Links
7.5
0
7.5
15
22.5
30 km

Copyright Transport for London (https://www.tfl.gov.uk/)

National Grid network derived from data from (2012) http://www2.nationalgrid.com/uk/services/land-and-development/planning-authority/shape-files/

Figure 5 – QGIS linked view of the two networks and their spatial dependencies shown in **Figure 2**.

As the encoding of a network or set of interdependent networks contains both topographical and topological information a network visualisation and analysis tool has been developed for the database schema using JavaScript Object Notation (JSON) data format that allows both views (topographic and topological) along with analytical metrics to be viewed. **Figure 6** illustrates several components of this this visualisation tool. In the tool a user is able to select a particular network its style, and switch between the topological and topographical views. The topographical view is currently delivered using the OpenLayers web mapping library (OpenLayers, 2014a), and uses the OpenLayers Style Map (OpenLayers, 2014b) to encode styles for each network. The topological view was developed using the D3 library (Bostock, 2014), and provides currently a single force-directed layout (Bostock, 2012) to visualise each network. **Figure 6** also illustrates the metric view which is derived by using the functionality of NetworkX.



Networks

Spatial Data
Paris RATP

Node Vector Style: Paris RATP Nodes (By type)
Node Label: Display Node Labels (black text on white) □
NodeID
Edge Vector Style: Paris RATP Edges (By route)
Edge Label: Display Edge Labels (white text on black) □
EdgeID

Networks

Spatial Data
Paris RATP

Node Style: Use node attribute
Node Size: Degree Centrality
Scale by: 1000

Node Label: Display Node Labels ☑
id
Edge Style: Paris RATP Edges (By type)
Edge Thickness: Edge Betweenness Centrality
Scale by: 545

Edge Label: Display Edge Labels □
(Source, Target)

Degree Histogram

Node(s):

type = default ■
type = rer 🟥
type = metro 🟦
type = tram 🟩
type = ■

Edge(s):

type = default
Stroke Width: 2
type = rer
Stroke Width: 2
type = metro
Stroke Width: 2
type = tram
Stroke Width: 2
type =
Stroke Width: 2

Figure 6 – Example of views of the topographic, topological and metric based capabilities of the database schema analysis and visualisation tool. **Top:** RATP Rail Network, **Middle:** topological view, **Bottom:** metric view as a degree histogram.

4. Conclusion

This paper has highlighted the key components of a network database schema developed to facilitate NI network modelling and analysis within a network-of-networks paradigm. By using table inheritance available within PostgreSQL the database schema, coupled with Python wrapping modules, provide the platform through which to conduct infrastructure network modelling and analysis. To address the ability to model interdependencies between networks, a simple mechanism through which interdependencies between two networks can be represented is included within the schema. The utility of the schema has been highlighted through the local-scale assessment of faecal sludge management options via a road-based network analysis. Additionally a range of visualisation and analysis tools that have been developed that can directly interface with the network database schema, enabling rapid visual and analytical analysis of the spatial, topological and metric characteristics of complex spatial infrastructure networks.

5. Acknowledgements

The authors would like to acknowledge funding from the Engineering and Physical Sciences Research Council (EPSRC) grant EP/I01344X/1 to the Infrastructure Transitions Research Consortium (ITRC).

6. Biography

Mr David Alderson received the B.Sc. (Hons) Geographic Information Science from Newcastle University in 2005. He is a Research Assistant in GeoInformatics in the School of Civil Engineering and Geosciences at Newcastle University.

Dr Stuart Barr is Senior Lecturer in Geographic Information Science at Newcastle University.

Dr Tomas Holderness is a Geomatics Research Fellow at the SMART Infrastructure Facility, University of Wollongong, Australia.

Mr Craig Robson is currently studying for a Ph.D. in spatial infrastructure network modelling at Newcastle University.

Mr Alistair Ford is a Researcher in Geomatics at the Newcastle University.

7. References

Bostock, M. (2012, 11 12). *Force-Directed Graph*. Retrieved 10 30, 2014, from mbostock's block #4062045: http://bl.ocks.org/mbostock/4062045

Kennedy-Walker, R., Holderness, T., Alderson, D., Evans, B., & Barr, S. (2014). Using crowd-sourced data for sanitation network modelling in informal settlements. *ICE Municipal Engineer , 167* (3), 157-165.

OpenLayers. (2014). *OpenLayers.StyleMap*. Retrieved 10 30, 2014, from OpenLayers JavaScript Mapping Library: http://dev.openlayers.org/releases/OpenLayers-2.13.1/doc/apidocs/files/OpenLayers/StyleMap-js.html#OpenLayers.StyleMap.OpenLayers.StyleMap

OpenLayers. (2014). *OpenLayers: Free Maps for the Web*. Retrieved 10 30, 2014, from OpenLayers 2: http://openlayers.org/two/

Quantum GIS. (2014). Retrieved 10 25, 2014, from Quantum GIS: http://www.qgis.org/en/site/forusers/download.html