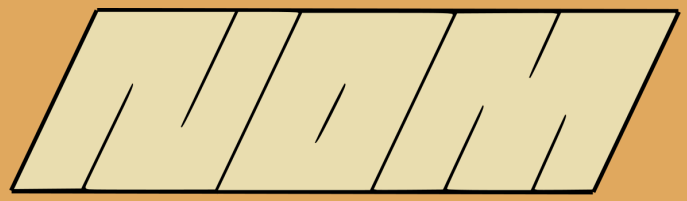
This project is funded and supported by:

THIS.Institute NIHR | Applied Research Collaboration South West Peninsula UNIVERSITY OF EXETER



Network-based Operational Modelling

Introduction to network analysis

Dr Sean Manzi - University of Exeter, UK

Health Service Modelling Associates Programme 2020

Session structure

- 0930 – 1000 Network analysis and its uses
- 1000 – 1030 Components of a network
- 1030 – 1045 Comfort break
- 1045 – 1130 Transforming data into a network representation
- 1130 – 1230 Building a network graph (NetworkX)

Learning objectives

- Understand what a network graph is and how it is constructed
- Be able to transform raw data into a format suitable for conducting a network analysis
- Know how to create a simple network graph in Python using NetworkX

Network analysis and its uses

What is a network

Network graphs are used to:

- Study the function of large and/or complex systems
- Visualise relationships and interactions in a intuitive way
- Represent abstract concepts in a concrete way

Exemplar uses

EXAMPLES OF NETWORKS AND THEIR COMPONENTS

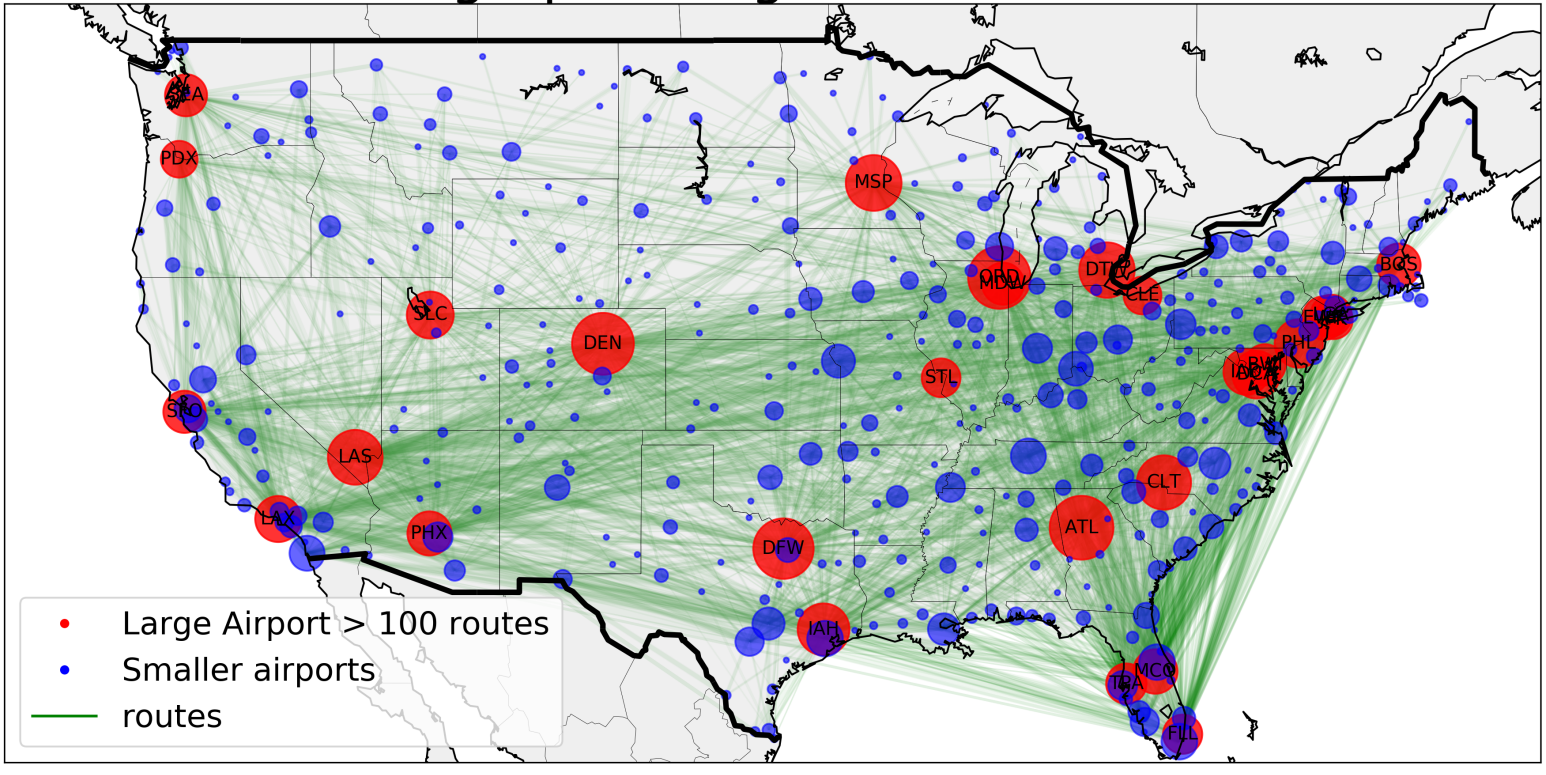
NETWORK	VERTICES	VERTEX ATTRIBUTES	EDGES	EDGE ATTRIBUTES
Airlines Network	**Airports**	Footfall, Terminals, Staff, City population, International/Domestic, Freight, Hangar capacity	**Airplanes / Routes**	Frequency, # Passengers, Plane Type, Fuel Usage, Distance covered, Empty seats
Banking Network	**Account Holders**	Name, demographics, KYC Document, Products, Account status, balance and other details	**Transactions**	Type, Amount, Authentication (pass/OTP), Time, Location, Device
Social Network	**Users**	Name, demographics, # connections, likes, circles belong to, subscriptions	**Interactions**	Medium (like/comment/direct message), time, duration, type of content, topic
Physician Network	**Doctors**	Demographics, speciality, experience, affiliation (type and size), Weekly patient intake	**Patients**	Demographics, Diagnosis history, visit frequency, purpose, referred to, insurance
Supply Chain Network	**Warehouses**	Location, size, capacity, storage type, connectivity, manual/automated	**Trucks**	Load capacity, # wheels, year of make, geographical permit, miles travelled. Maintenance cost, driver experience

TheMathCompany

Credit: https://www.analyticsvidhya.com/blog/2018/04/introduction-to-graph-theory-network-analysis-python-codes/

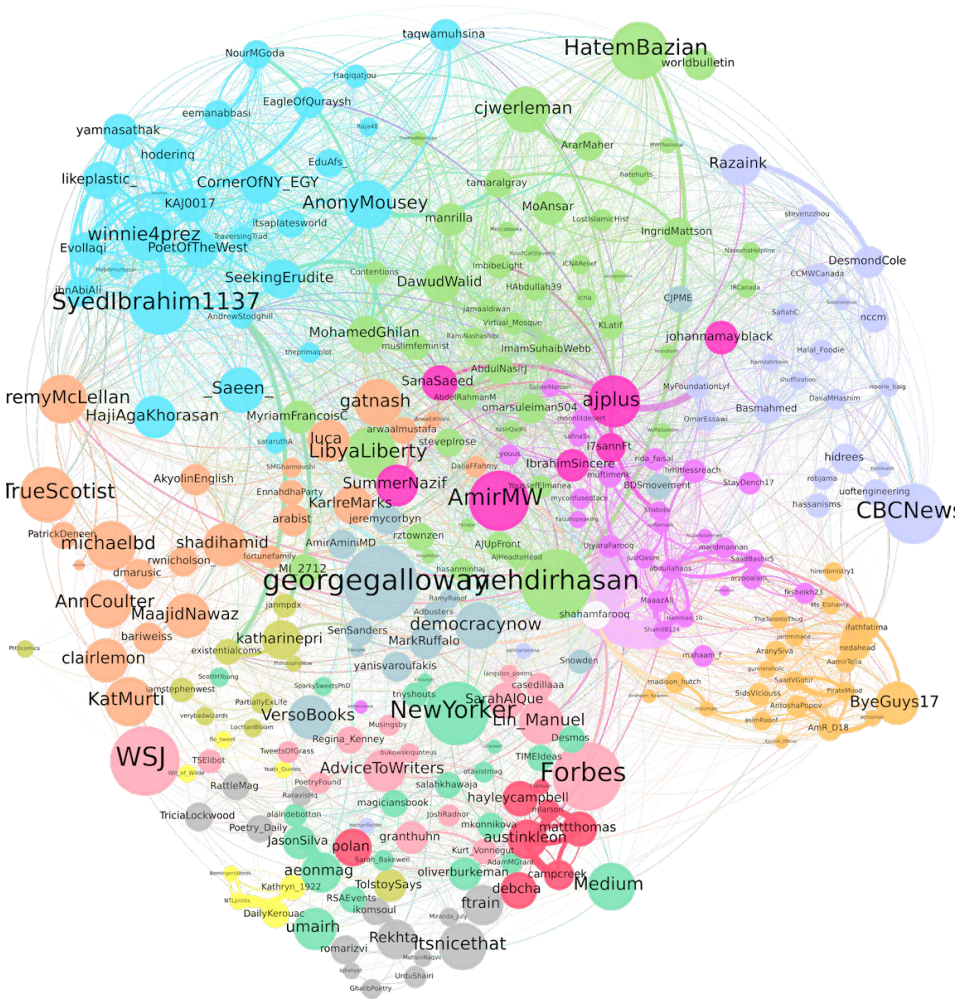
Example 1

Network graph of flight routes in the USA



Credit: towardsdatascience.com

Example 2



Credit: towardsdatascience.com

Network-based Operational Modelling

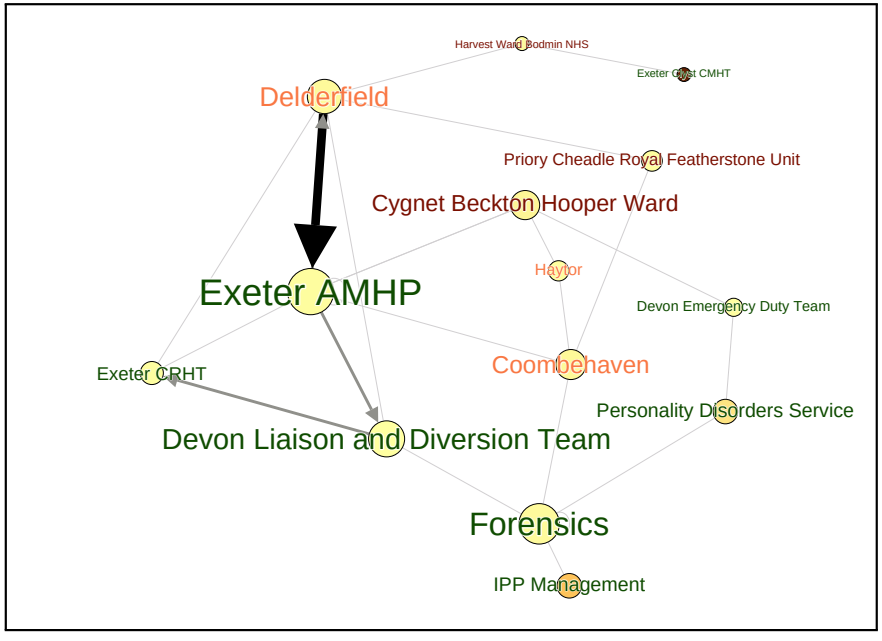
THIS.Institute

NIHR | Applied Research Collaboration South West Peninsula

University of Exeter

Network-based operational modelling for healthcare

- Graph as service structure
- Node as service
- Edge as patient movement
- Edge weight as number of patients
- Node attributes as improvement measures



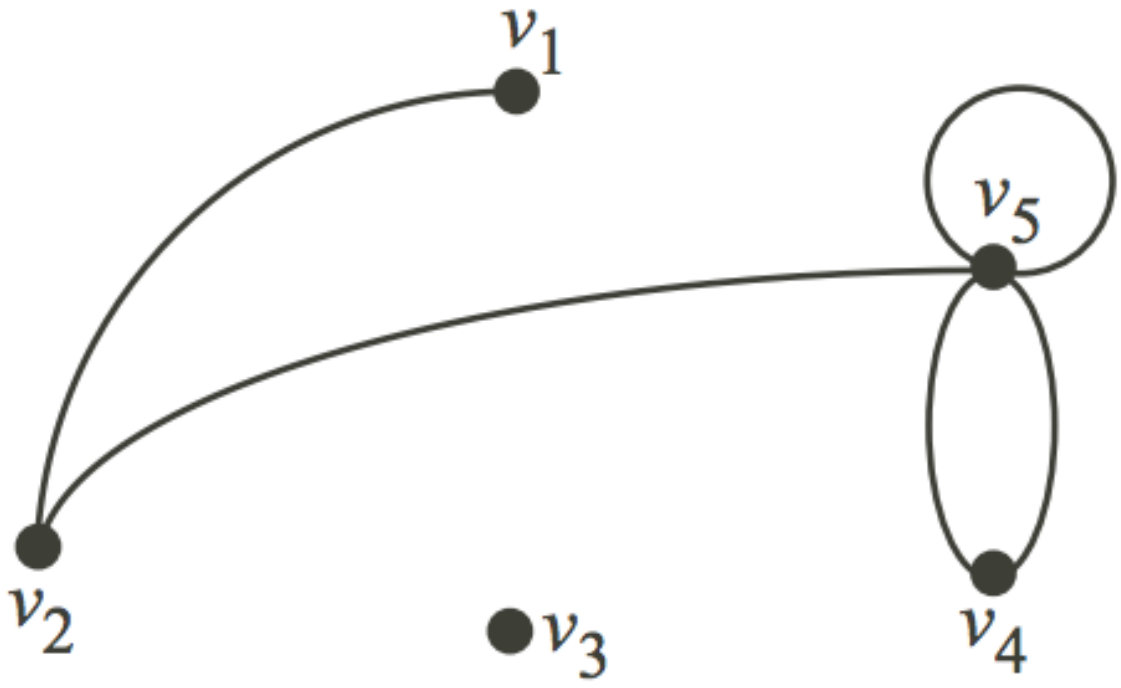
Components of a network

What is a network

A network graph is:

- A graph based representation of a system
- Based on pairwise relationships between entities
- Quantitatively encodes the properties of the system within the network structure

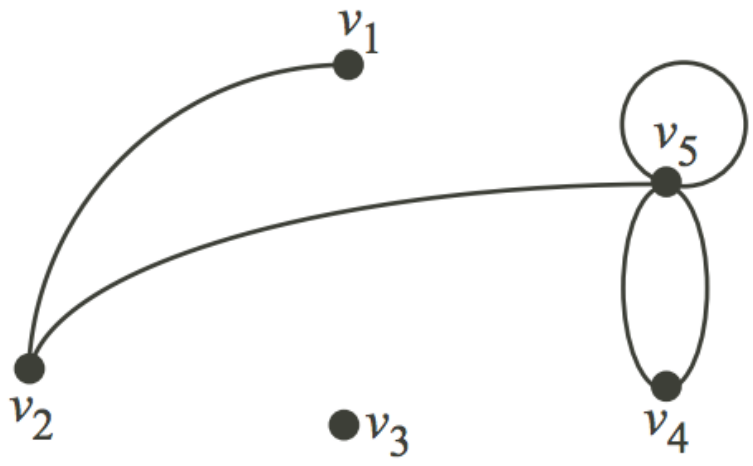
Components of a network graph



Credit: https://www.analyticsvidhya.com/blog/2018/04/introduction-to-graph-theory-network-analysis-python-codes/

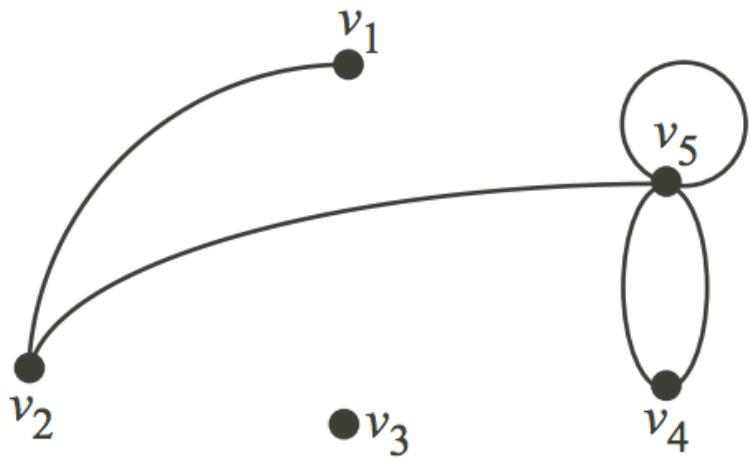
Components of a network graph

- Node/vertex
- Edge
- Graph - unordered pairs of nodes
- DiGraph - ordered pairs of nodes
- Edge weight
- Node attributes



Some terminology

- Parallel edges
- Loop
- Adjacent edges
- Adjacent nodes



Required input data

Data required to build a network graph

- Node list: ID, Label, n-attributes
- Edge list: Source, Target, n-attributes (e.g. Weight)

Break time

Break time!!

When we return we will look at how we create node and edge data from raw data tables

Data transformation

Data transformation process

1. Start with the raw data table - patient level data
2. Clean the data - category consistency, negative dates, erroneous whitespace
3. Order the data - by patient then by date
4. Create an adjacency matrix - more on this shortly
5. Use unique instances as nodes - depends on the resolution of the data and how you want to subset your data
6. Iterate over the interaction matrix to create edge list - source, target, weight

Data cleaning

Some things to consider

- Capitalisation inconsistencies
- Spelling mistakes
- Erroneous whitespace - leading, trailing, multiples
- Incorrect date entries - reversed, missing
- Missing data

Adjacency matrix

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	1	1	1	0	1	0	0	0
3	0	0	1	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0
5	0	0	1	1	0	9	2	0	0
6	0	0	0	0	0	2	1	0	0
7	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0
12	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	5	3	0	0

- Source in row, target in column, frequency count in cell

Node list

Index	ID	Label	MeanLoS	MedianLoS	Setting
0	0	All OOA services	30.4444	26	OOA
1	1	Area 10 AMHP	23	23	Community
2	2	Area 10 CRHT	4.625	3	Community
3	3	Area 10 Community Mental Health	1286.5	176.5	Community
4	4	Area 10 OPMH Community	505	234	Community
5	5	Area 11 MHAT	81.1143	58	Community
6	6	Area 12 CMHT	793.818	300	Community
7	7	Area 13 MHAR	1909.33	928	Community
8	8	Area 14 CMHT	550	550	Community
9	9	Area 15 CMHT	227.5	227.5	Community
10	10	Area 15 OPMH Community	173.429	85	Community
11	11	Area 16 Community Mental Health	449.5	234.5	Community
12	12	Area 17 CMHT	764	764	Community
13	13	Area 18 PPT	421	294	Community
14	14	Area 19 CMHT	1090	1090	Community
15	15	Area 20 OPMH Community	241	241	Community

- Required: ID
- Optional: Label, continuous and discrete attributes

Edge list

Index	Source	Target	Type	Id	Weight
0	37	0	Directed	0	1
1	49	0	Directed	1	1
2	57	0	Directed	2	1
3	58	0	Directed	3	1
4	59	0	Directed	4	1
5	69	0	Directed	5	1
6	73	0	Directed	6	1
7	83	0	Directed	7	1
8	2	1	Directed	8	1
9	2	2	Directed	9	1
10	3	2	Directed	10	1
11	5	2	Directed	11	1
12	47	2	Directed	12	1
13	49	2	Directed	13	1
14	56	2	Directed	14	1
15	86	2	Directed	15	1

- Required: Source, Target, ID
- Optional: Type (required for Gephi), weight, continuous and discrete attributes

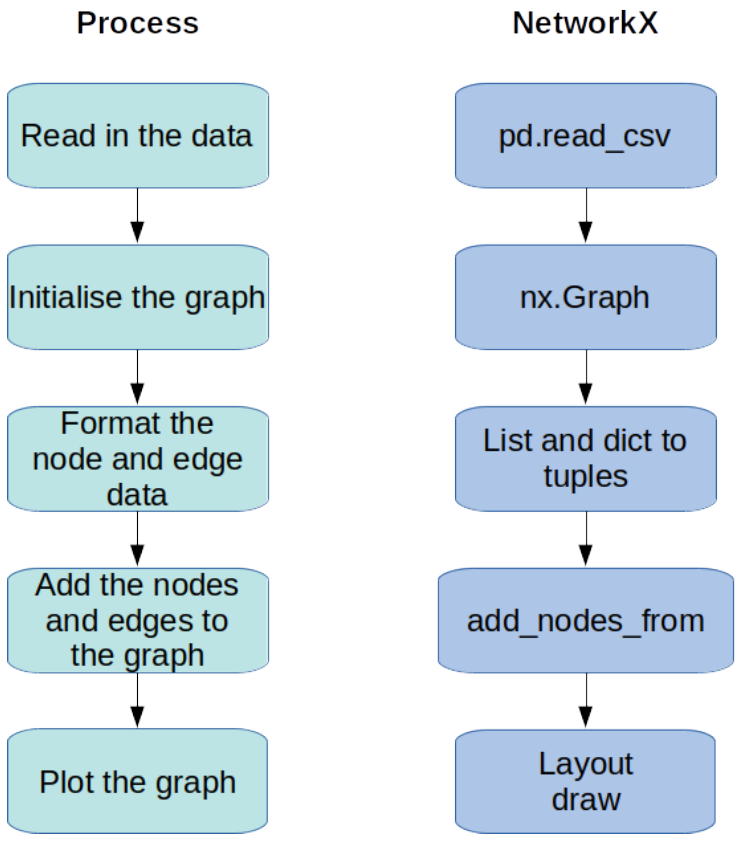
Task time

Data transformation

- Open the data_transformation.py file in spyder
- This is an example of how to transform patient episode data for network analysis
- This example provides all of the required columns for use with multiple network analysis libraries
- Set your working directory
- Run the code line by line to determine what is happening
- Insert print commands and/or use the variable explorer to examine the data transformations

Building a network graph

Building a network graph



Task time

Open Spyder and follow along, typing in the code and running it as we go

Building a network graph

Import the required libraries and the data we just created

```
import networkx as nx
import pandas as pd

nodes = pd.read_csv('data/node_list.csv',
low_memory=False)
edges = pd.read_csv('data/edge_list.csv',
low_memory=False)
```

Building a network graph

Initialise a graph object which we will assign to the variable **G**

```
G = nx.Graph()
```

Building a network graph

Node inputs are tuples of the node ID and a dictionary of attributes
Node tuples (ID, dict of attributes)

```
idList = nodes['ID'].tolist()
labels =  pd.DataFrame(nodes['Label'])
labelDicts = labels.to_dict(orient='records')
nodeTuples = [tuple(r) for r in zip(idList,labelDicts)]
```

Building a network graph

Edge inputs are tuples of source ID, target ID and a dictionary of attributes
Edge tuples (Source, Target, dict of attributes)

```
sourceList = edges['Source'].tolist()
targetList = edges['Target'].tolist()
weights = pd.DataFrame(edges['Weight'])
weightDicts = weights.to_dict(orient='records')
edgeTuples =
[tuple(r) for r in zip(sourceList,targetList,weightDicts)]
```

Building a network graph

We add the nodes and edges to the graph object **G**

```
G.add_nodes_from(nodeTuples)
G.add_edges_from(edgeTuples)
```

Building a network graph

We pass the graph object to a layout algorithm to determine the node coordinates and hence the edge start and end points

```
pos = nx.kamada_kawai_layout(G)
```

Building a network graph

Finally we draw the graph passing in the graph object **G**, the position object **pos** and adjust the node size

```
plot = nx.draw(G, pos=pos, node_size=100,
 with_labels=True, font_size=10)
```

Advanced network analysis

In the optional advanced sessions we will cover:

- Different types of graph
- Graph metrics and their interpretation
- How graph visualisations work
- Graph algorithms and custom layouts
- Interactive graphs with Plotly and Holoviews

Thank you

Thank you for paying attention
Hope you enjoyed the session
Checkout https://www.project-nom.com for more information and training on the use of network-based operational modelling for whole system modelling in healthcare