TECHNION – Israel Institute of Technology
The William Davidson Faculty of Industrial Engineering & Management

Center for Service Enterprise Engineering (SEE)
https://seelab.net.technion.ac.il/

SEEGraph Viewer 2.0

Lesson 2: Understanding the Graphs

Created: May 22, 2016
Last Updated: February 1, 2021

SEEGraph files root folder: C:\animation

SEELab software applications include a variety of graphs-related algorithms that support graph creation, EDA, and layout algorithms, which are required for graph and network drawing. Specifically, these includes

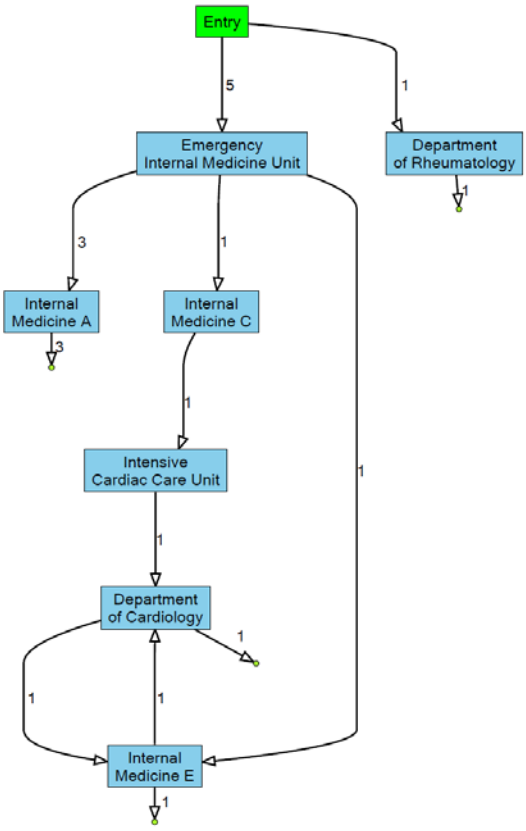
- *SEEGraph* (Designer), that supports mining of graph-structure (sub-structures) from databases. This is a prerequisite for the next phase, which is carried out by
- *SEEGraph Viewer*, which visualizes graphs and data-animations; the prerequisite for visualization is the graph-layout algorithm.

The main steps in the process of creating graph: to this end, a simple example of "patients flow in a hospital" is provided in Appendix 3. Here we just list the steps in this creation process:

1. Preparing Data (input to design part of SEEGraph). This constitutes around 90% of the total work.
2. Designing Graph (via the SEEGraph software). This entails characterization of graph attributes and structure.
3. Displaying Graph or Network (via the SEEGraph Viewer software), by layout method.

The "patient flow" example is characterized by the following table (left) which, when fed to SEEGraph Viewer, and combined with user-specified graph-attributes (e.g. colors, fonts) and graph-layout (below it is hierarchical), produces the graph on the right (as well as may others, per user demand – see Appendix 3).

patient_id	entry_date	department
1	02/01/2005 16:38	Emergency Internal Medicine Unit
1	02/01/2005 23:55	Internal Medicine A
2	04/01/2005 14:07	Emergency Internal Medicine Unit
2	04/01/2005 20:23	Internal Medicine A
3	01/01/2005 15:58	Emergency Internal Medicine Unit
3	01/01/2005 17:42	Internal Medicine A
4	06/01/2005 09:35	Emergency Internal Medicine Unit
4	06/01/2005 16:13	Internal Medicine E
4	10/01/2005 08:00	Department of Cardiology
4	11/01/2005 08:00	Internal Medicine E
5	02/01/2005 14:56	Emergency Internal Medicine Unit
5	02/01/2005 21:19	Internal Medicine C
5	07/01/2005 16:37	Intensive Cardiac Care Unit
5	11/01/2005 13:31	Department of Cardiology
6	30/01/2005 08:44	Department of Rheumatology



SEEGraph Viewer 2.0

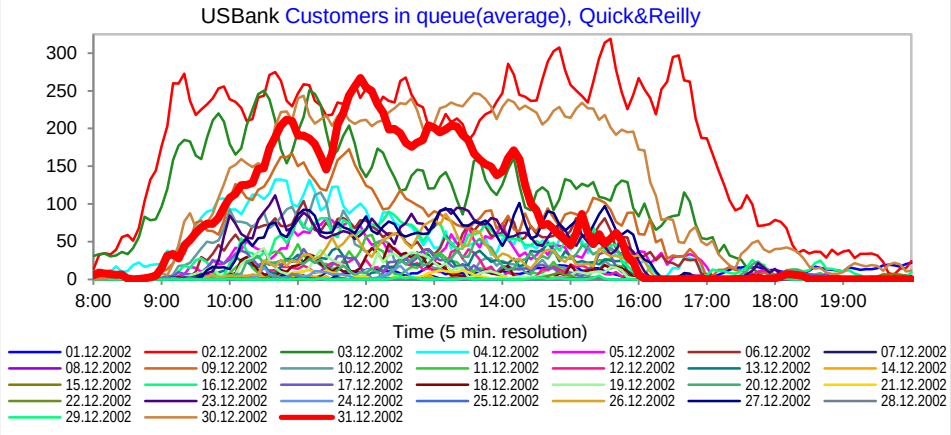
Contents

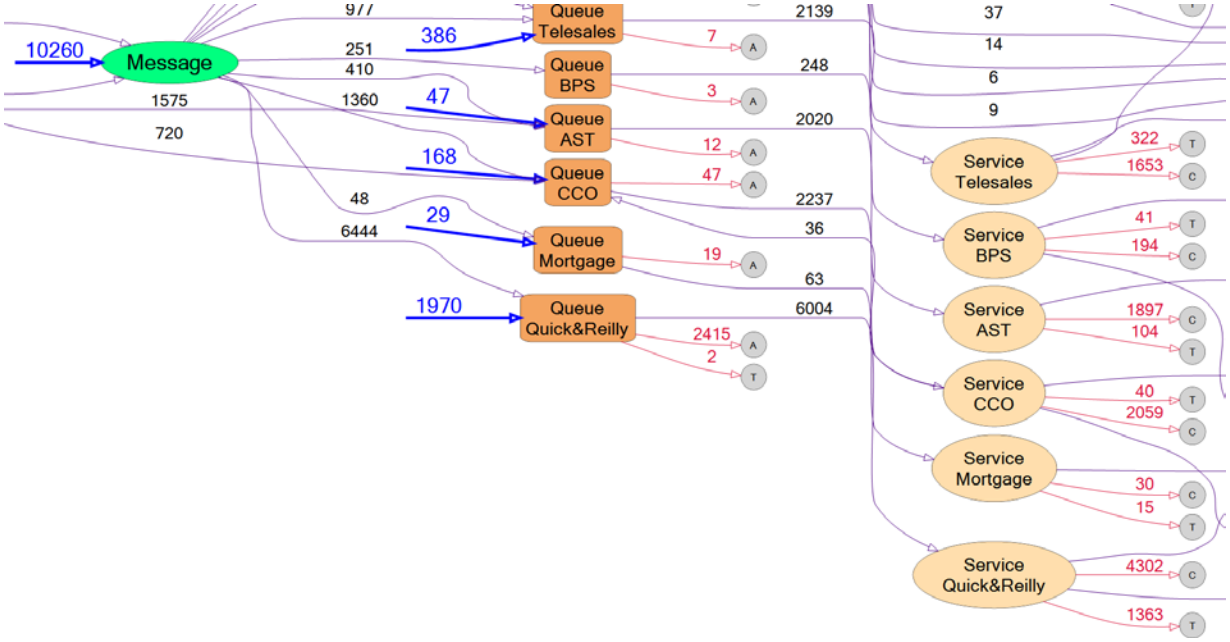
1. Graphs (Static)

1.1 Daily Totals: customer calls (USBank Call Center)

Data sample - December 31, 2002 (SEEGraph file: root folder \cust_subcalls_trn_l2.txt[1])

Remark: December 31 will be used throughout the tutorial. The reason is that, as far as services related to stocks and bonds, this is one of the heavy-traffic days (being the end of the financial year). See, for example, the following Quick&Reilly service type (31.12.2002 is **bold-red**):

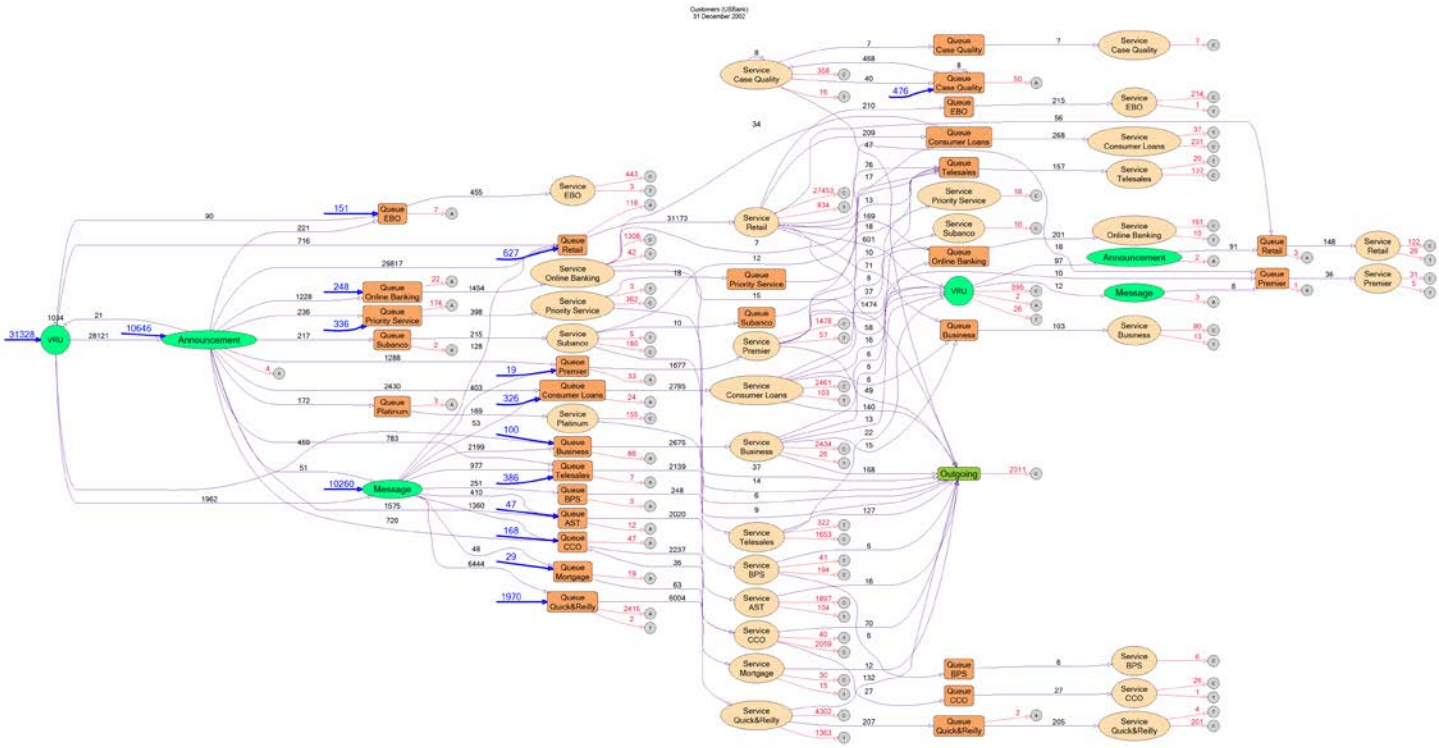




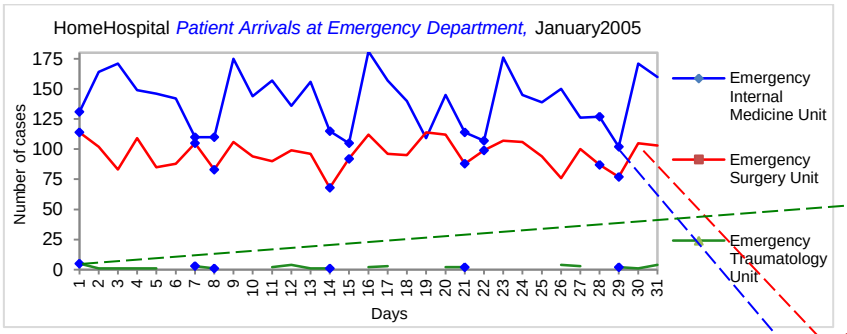
Exercise: Via SEEStat, reproduce a daily summary of call center activity, similar to one in the graph above, in terms of the following measures: arrivals to queue, arrivals to service (departure from queue), abandons.
(In SEEStat: Time series-> Daily totals-> select variable-> select category-> select month (*December 2002*)-> Days for one month, then focus on December 31, 2002,).

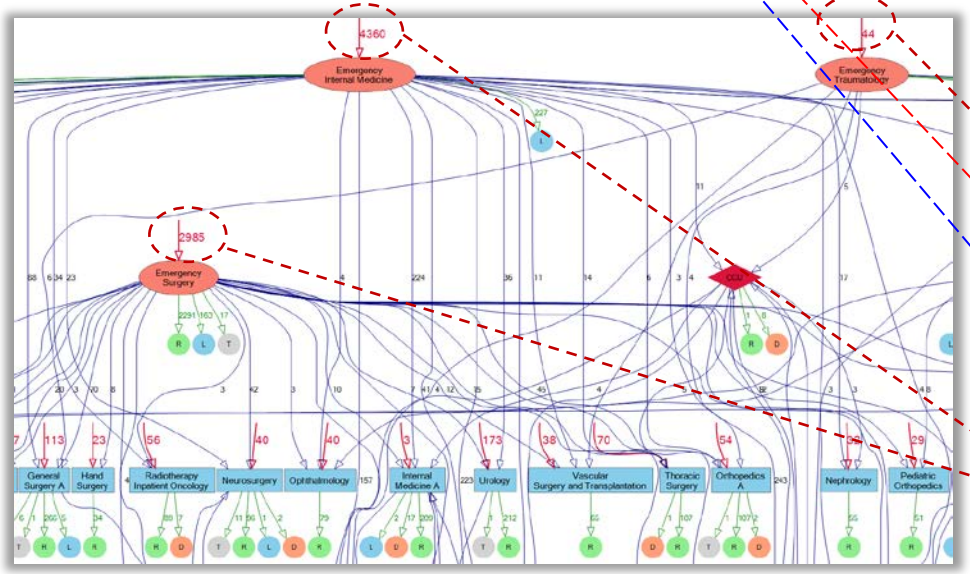
[1] To run SEEGraph Viewer, one must open a text-file that includes a graph description – in the above, *root folder\cust_subcalls_trn_l2.txt is the name of such a file. We are keeping these names in the document, to facilitate future archiving if needed.* .

To see the graph clearly, click here.



1.2 Monthly Totals: patient arrivals at hospital/discharges from hospital (HomeHospital)



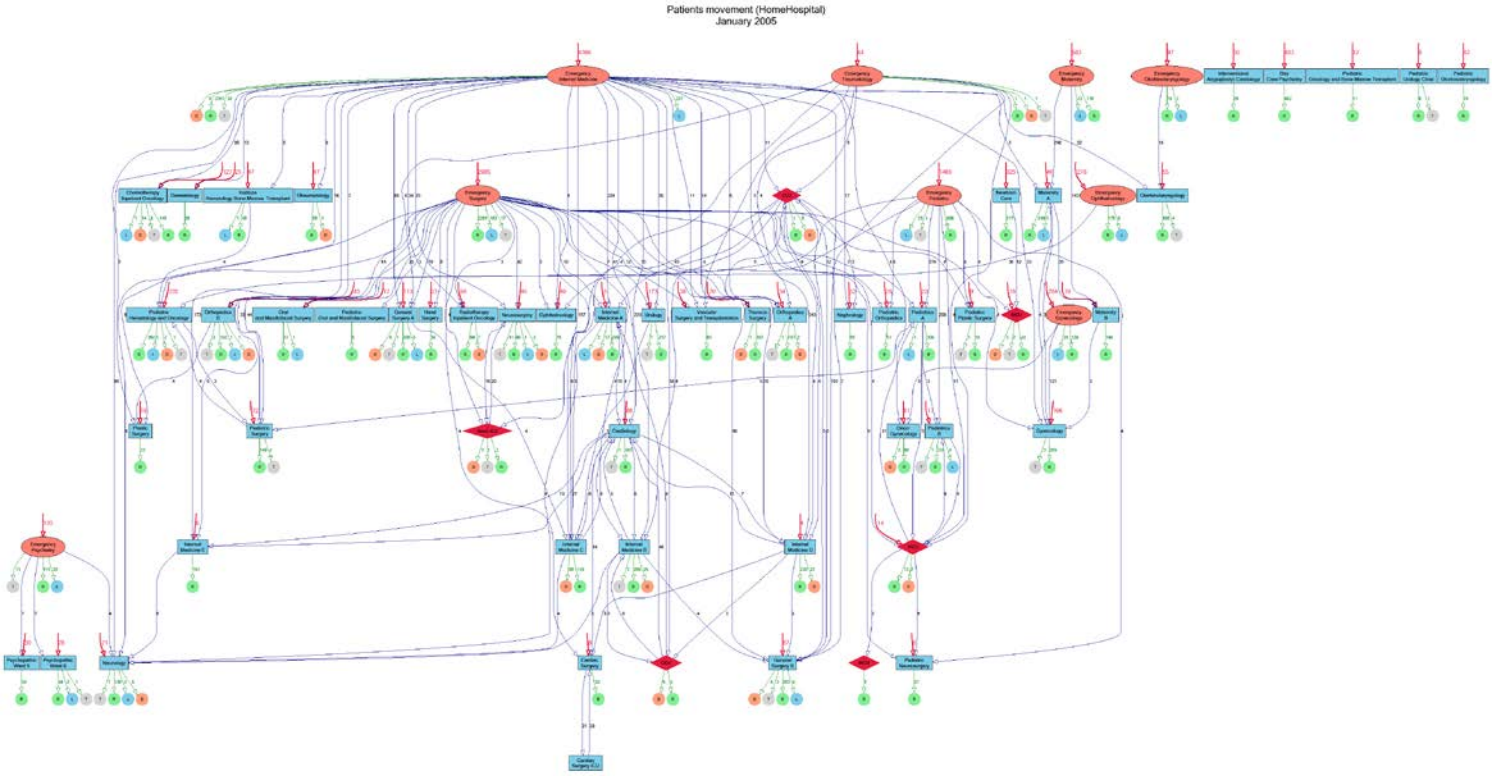


Time Series, HomeHospital, Patient Arrivals at Emergency Department, January 2005			
Date	*Emergency Internal Medicine Unit*	*Emergency Surgery Unit*	*Emergency Traumatology Unit*
1	131	114	5
2	164	102	1
3	171	83	1
4	149	109	1
5	146	85	1
6	142	88	
7	110	105	3
8	110	83	1
9	175	106	
10	144	94	
11	157	90	2
12	136	99	4
13	156	96	1
14	115	68	1
15	105	92	
16	181	112	2
17	157	96	3
18	140	95	
19	109	114	
20	145	112	2
21	114	88	2
22	107	99	
23	176	107	
24	145	106	
25	139	94	
26	150	76	4
27	126	100	3
28	127	87	
29	102	77	2
30	171	105	1
31	160	103	4
monthly totals	4360	2985	44

Remark: The above was created separately in SEEStat (Figure and Table) and in SEEGraph (Graph, or Animation; the full one appears in the next page); then connections among the three (Table, Figure, Graph) were made manually.
It is clearly desirable to be able to produce all of the above at a single push-of-a-button plus few mouse-drags. This is conceptually and technically feasible (perhaps even easily), and it would have been natural for SEELab to do, had it been either a commercial venture or be blessed with ample resources. Since neither apply, efforts had to be prioritized and, and we opted to direct them to support research (as opposed to perfecting the existing, for example user-interface).

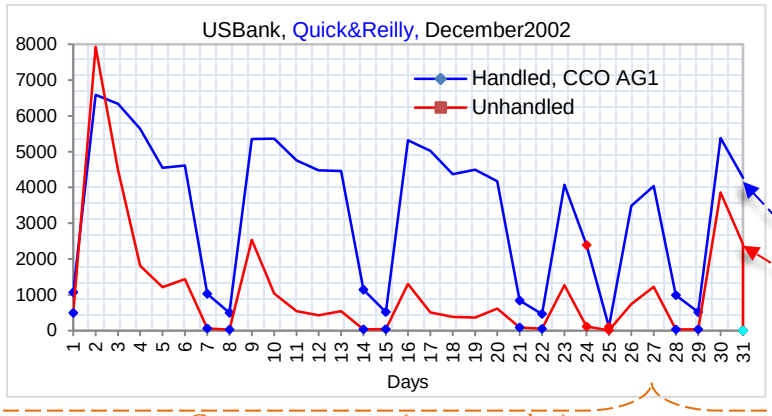
Data sample - January, 2005 (SEEGraph file: root folder \Patients_HomeHospital.txt)

To see the graph clearly, click here.

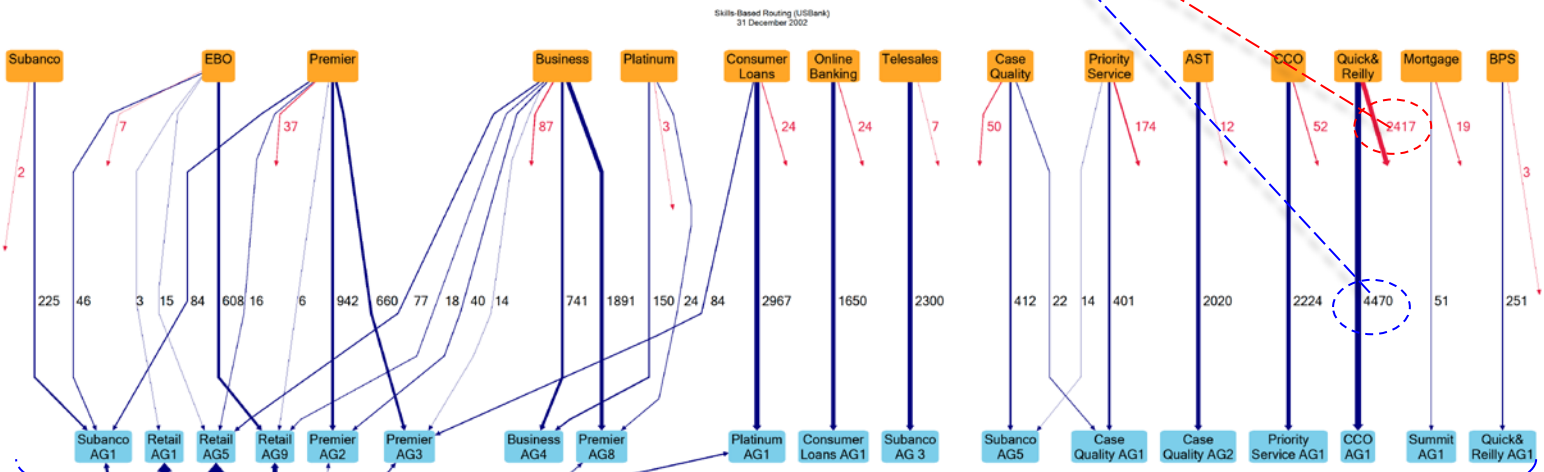


1.3 Bipartite Graph (daily totals): Skills-Based Routing (SBR) (USBank Call Center)

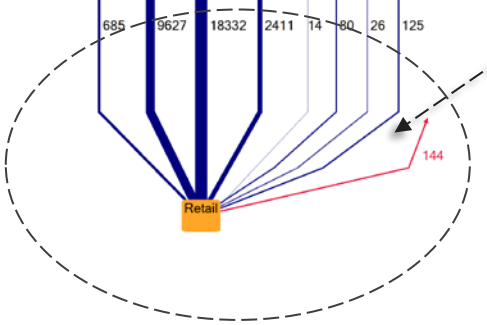
Data sample - December 31, 2002 (SEEGraph file: root folder \SBR_sinks.txt)



Customer queues (orange color)



Agent skill groups (blue color)



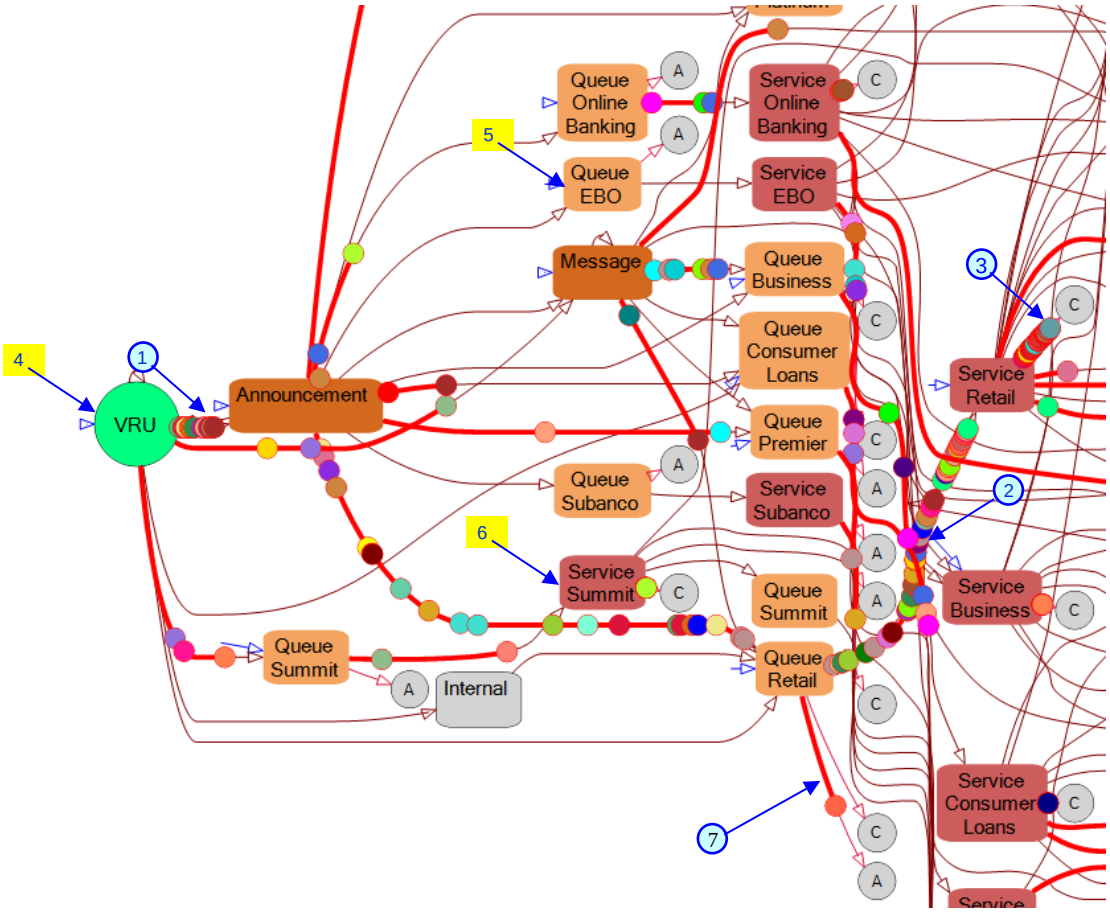
Balance by main left: most connected components move below.

Other types of balance:

- *No balance – one set (queues) on the one side, the other set (service pools) on the other side.*
- *By left nodes – arrange queues in an event manner*
- *By left node groups – arrange the graph so that the queues that connect to the same service pools on one side*

2. Process Views (Structure) / Data Animations

2.1 The Process Mining (Flow) view (USBank Call Center)



queue-start event
service-start event

Circles – correspond to customers within the call center.
Arcs – correspond to precedence relations between events.
Motion – constant speed corresponds to the time-between-events (e.g. spent in queue/service/IVR): the shorter the time the faster the motion. (Note: speed depends of course also on arc-length which, in turn, is determined by aesthetic considerations.)

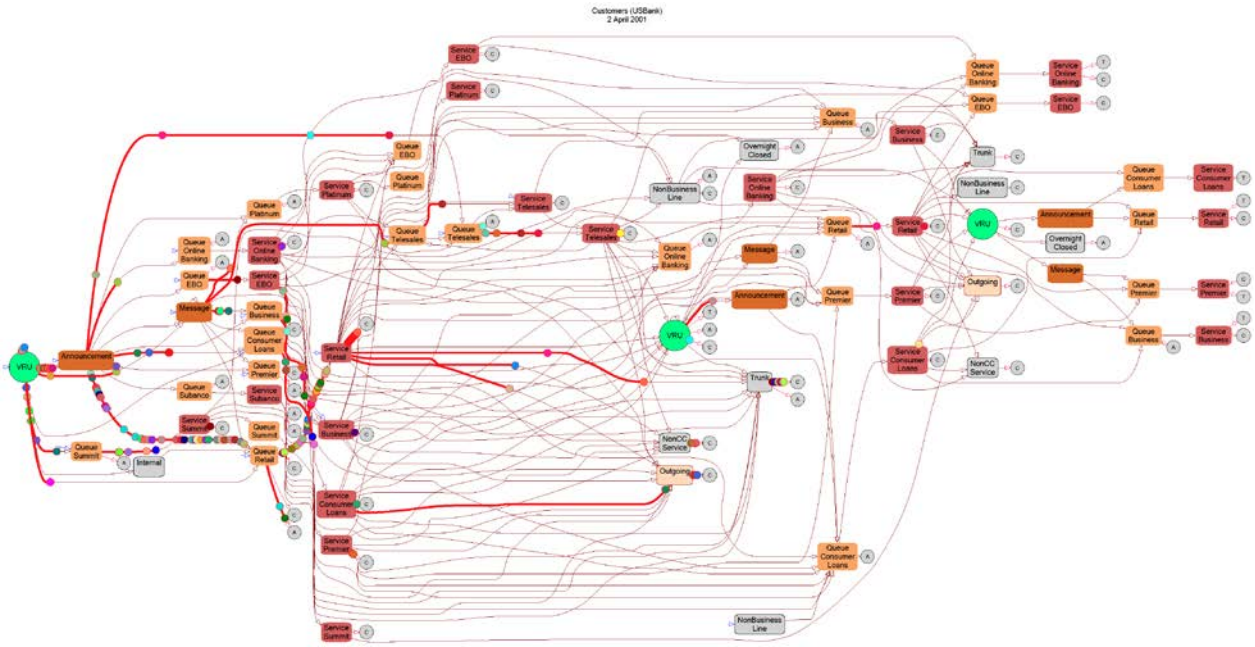
Note: Recall that PERT/CPM diagrams can be drawn via "Activity-on-Arc" vs. "Activity-in-Node". We do have the latter option in SEEStat, but it is boring visually; moreover, its advantages will be captured by the "hybrid" view, which will present in the sequel.

(1) On arc: customers in the VRU/IVR. Duration/speed on arc corresponds to IVR service time.
(2) On arc: Retail-customers in queue for service. Duration/speed on arc corresponds to WAIT time.
(3) On arc: Retail-customer in service. Duration on arc corresponds to SERVICE time. Note that the arc ends with a **C**, which stands for service **C**ompletion.
(4) Node: event VRU-start-time (start-event). (In the ACD data, this is the first customer sub-call.)

(5) Node: event Queue-entry of EBO customer (entry-event). Customers either enter this queue from either the VRU/Announcement or externally; the latter would also be the first sub-call.
(6) Node: event Service-entry of Summit customer (entry-event); this is again the first sub-call.
(7) On arc: customers who **A**bandon the queue (hence **A**). Duration on arc corresponds to WAIT time before abandonment; this is the first sub call.

Data sample - April 2, 2002 (SEEGraph file: root folder \cust_subcalls_trn_PM.txt)

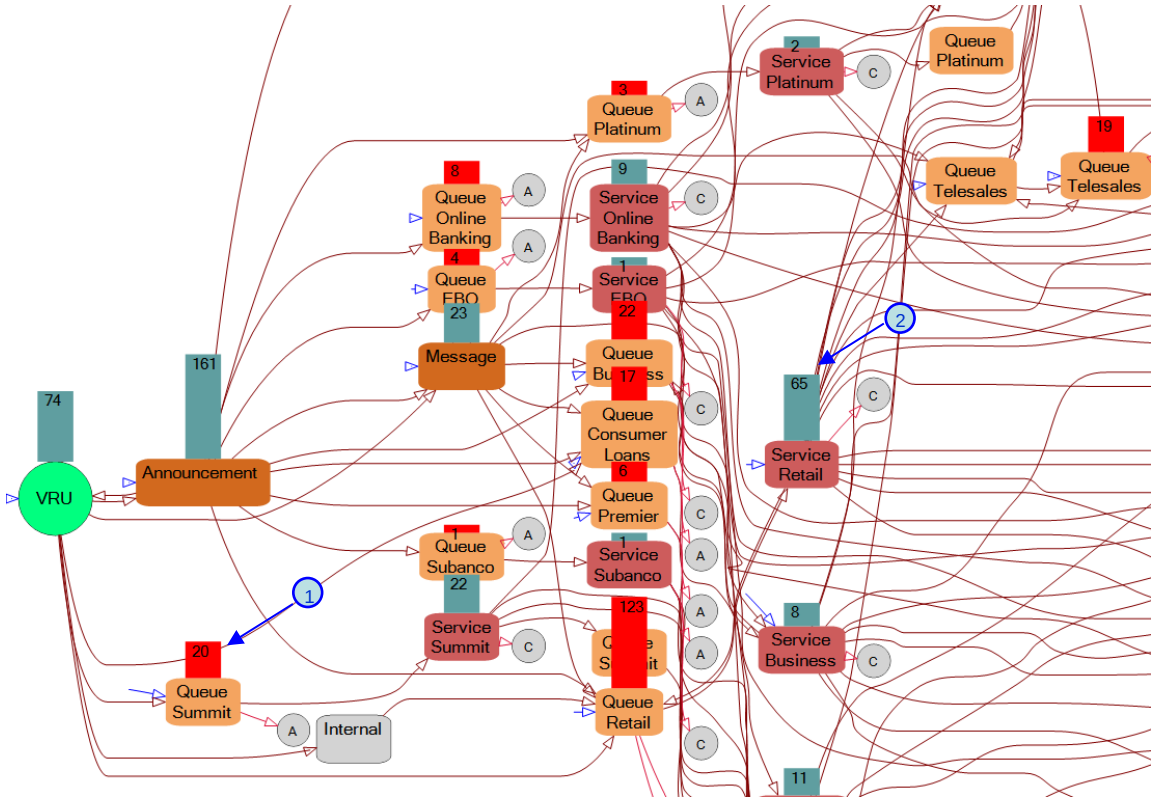
To see the graph clearly, click here.



2.2 The Operations-Research (or Queueing) view

2.2.1 Call-Center Queues (USBank Call Center)

Data sample - April 2, 2002 *(SEEGraph file: root folder \cust_subcalls_trn_OR.txt)*

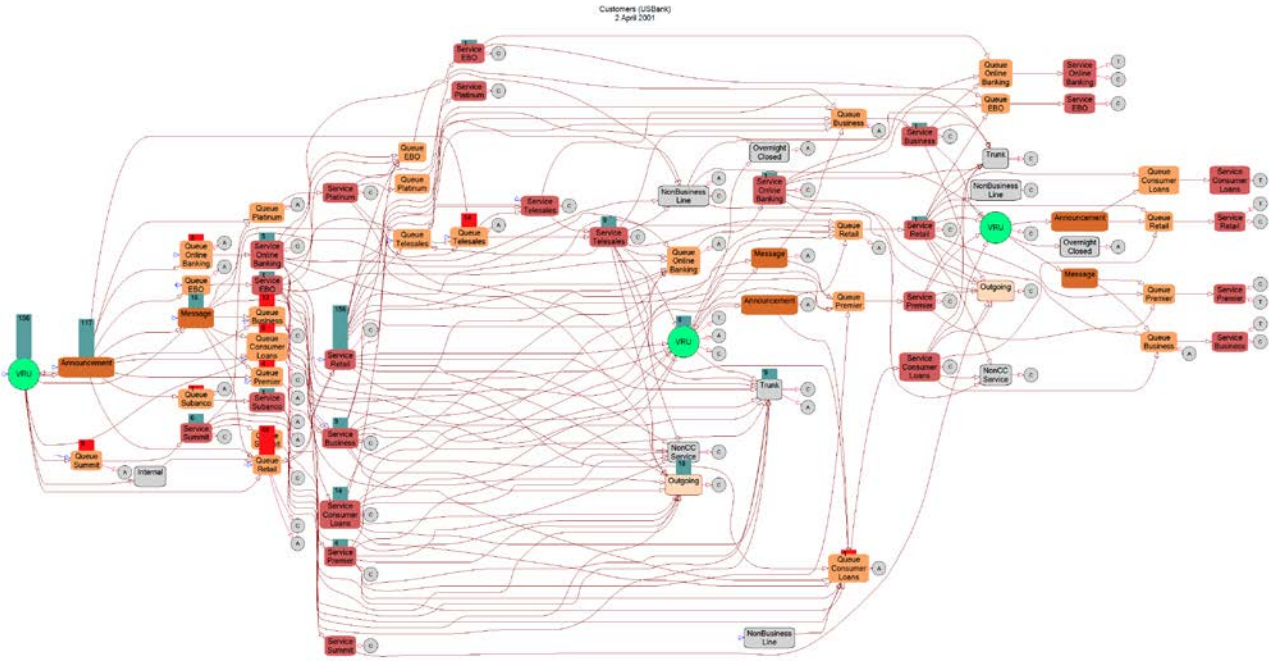


(1) Number of customers waiting in the Summit queue (20 customers above). Red color indicates that the maximal waiting time exceeds 60 seconds (a threshold that can be changed).
(2) Number of Retail customers that are in the process of receiving service (65 customers above).

Exercise:

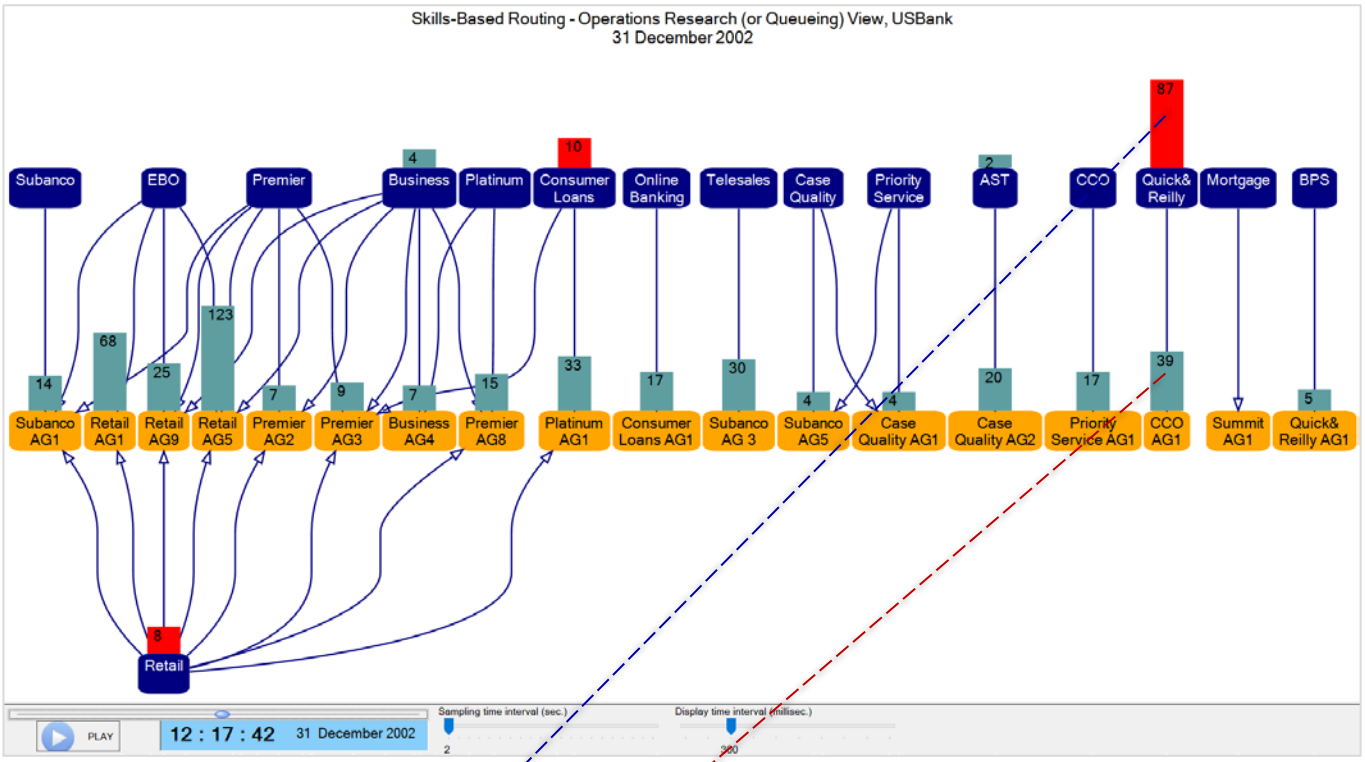
1. How many events are there between start-time and end-time.
2. Reproduce the information that appears in the graph above via SEEStat (number of customers in queue/number of customers in service during a day. Do not cover the VRU.

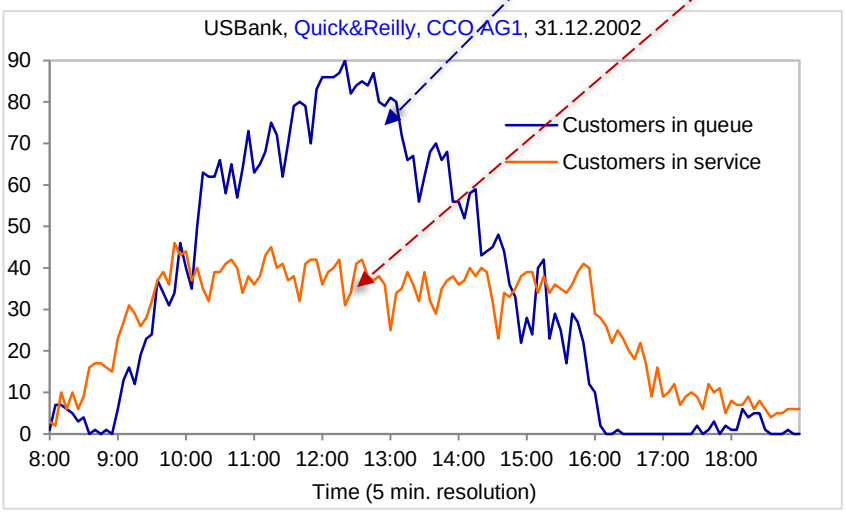
To see the graph clearly, click here.



2.2.2 Bipartite Graph: Skills-Based Routing (SBR) (USBank Call Center)

Data sample - December 31, 2002 (SEEGraph file: root folder[2] \ SBR_sinksOR.txt)





Continued next page

Bipartite graphs naturally capture relations (via arcs) between elements in two disjoint groups (nodes). Here elements in the groups are agent-pools and customer-queues. Arc direction trace the (virtual) motion of customers (e.g. from queue to service).

[2] *See SEEGraph files root folder in page 1*

SBR Research: The above SEEnimations depict dynamics of, what can be called, a Skills-Based-Routing (SBR) network. The terminology arose from call centers, where SBR is the algorithm that matches customers (needs) with servers (capabilities). An elementary teaching-note on SBR, old (from 2000, written before SBR became an important research-subject in Applied Probability and Stochastic Control) is the following, by Garnett and M.:
An Introduction to Skills-Based Routing and its Operational Complexities
Here is an accompanying lecture (also old and relevant).

As a stochastic network, SBR networks are called *parallel-server systems.* A pioneering recommended paper on such systems, in conventional heavy-traffic, is
Heavy traffic resource pooling in parallel-server systems, by Harrison & Lopez (1999).

The (asymptotically) optimal-control of a parallel-server system in conventional heavy-traffic (each server pool has a single- or mostly few servers in heavy-traffic) is established in
Scheduling Flexible Servers with Convex Delay Costs: Heavy-Traffic Optimality of the Generalized c-mue Rule, by M. and Stolyar (2004).

Advances for many-server pools, showcasing state-space-collapse, were achieved is
Simplified Control Problems for Multiclass Many-Server Queueing Systems, by Atar, M. and Shaikhet (2009)

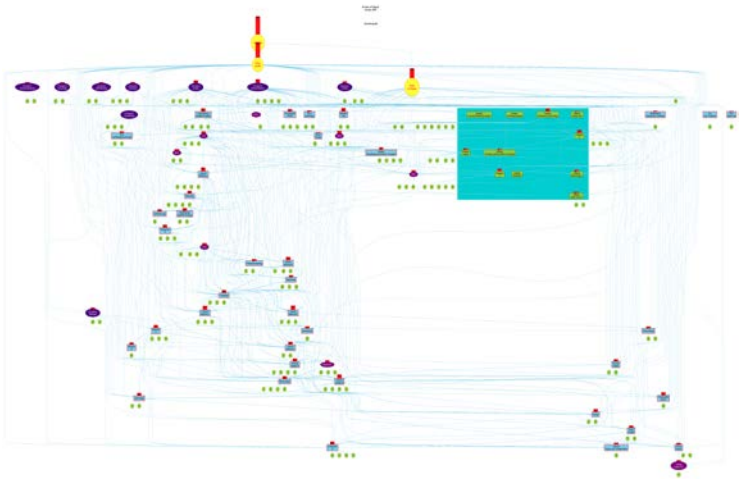
Finally, a recent paper where one can learn about the state-of-the-art is
Dynamic Scheduling in a Many-Server Multi-Class System: the Role of Customer Impatience in Large Systems, by Kim, Randhawa and Ward (2017).
This paper analyzes QED-SBR in a queueing system with multiple classes of customers and a single pool of servers (multi-class M/M/N+GI).

2.2.3 Number of patients in hospital/ED/wards (HomeHospital)

Data sample – all records of patients who were in hospital sometime during January 2005
(SEEGraph file: root folder \Patients_HomeHospital_OR.txt Please wait until the graph opens.

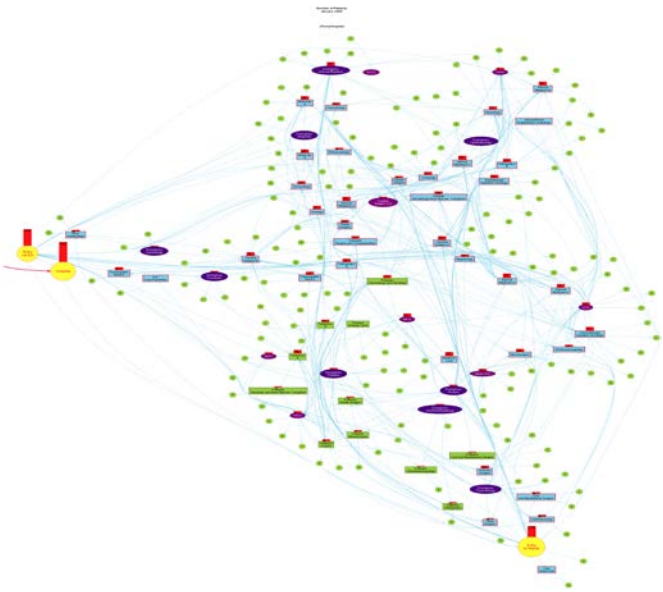
Hierarchical Layout (SEEGraph Viewer 1.0;
The other 4 layouts are available in Viewer 2.0)

To see the graph clearly, click here



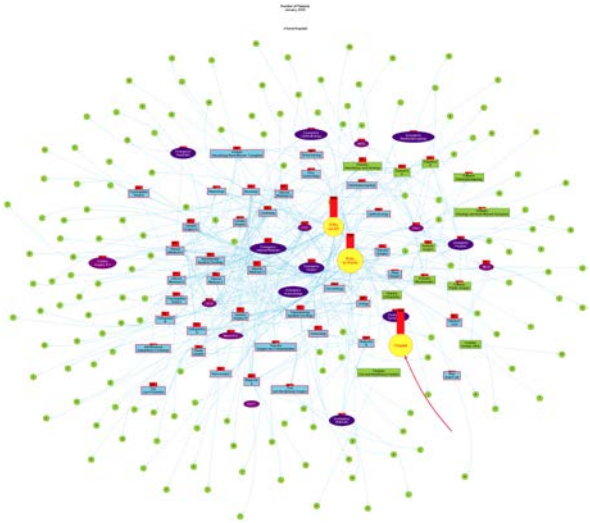
Circular Layout

To see the graph clearly, click here



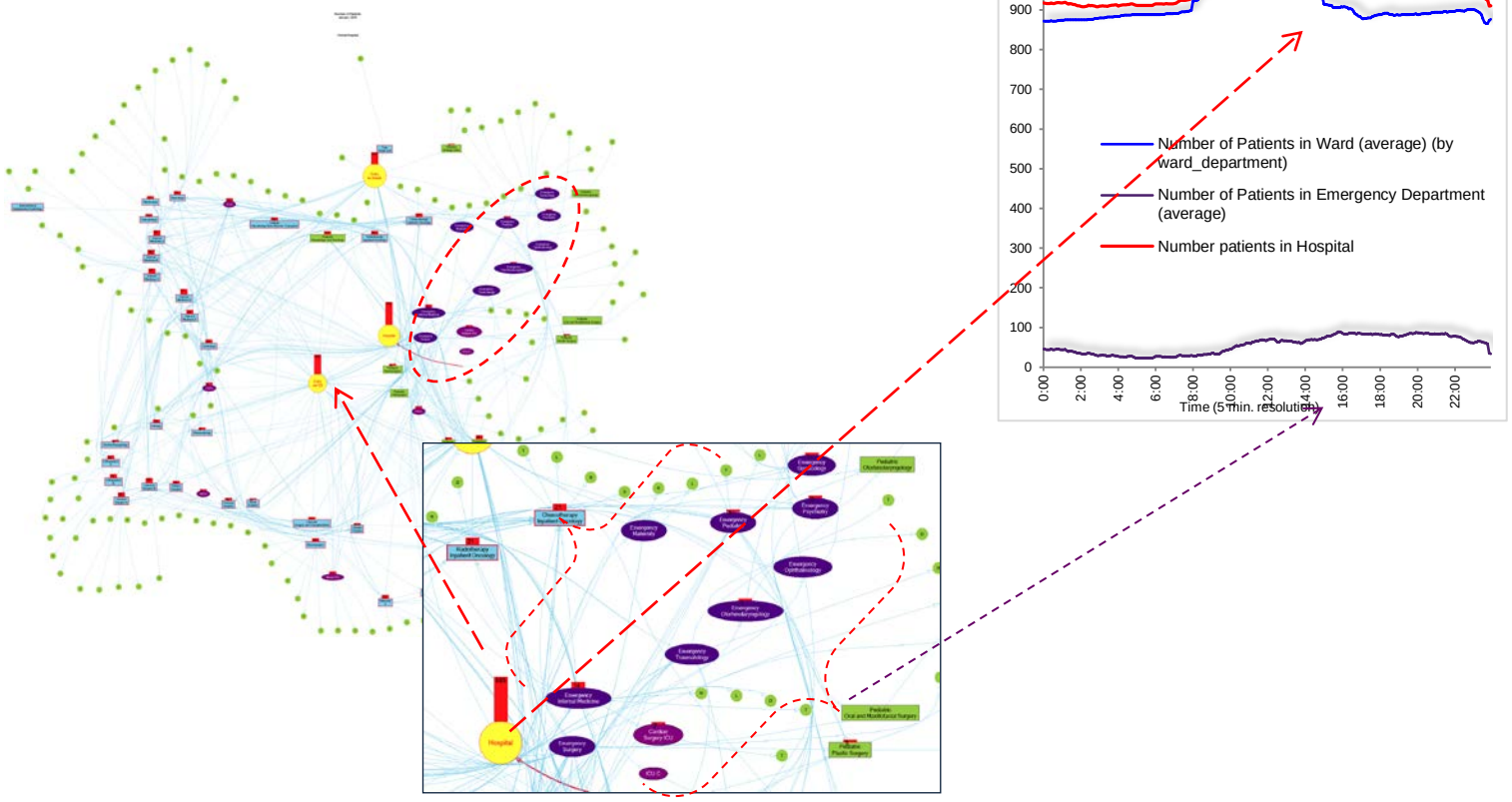
SFDP Layout

To see the graph clearly, click here

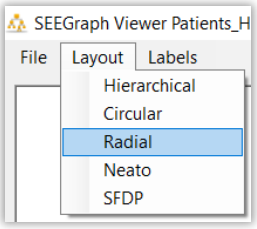


Radial Layout

To see the graph clearly, click here



Select *Layout*. Please wait until the graph opens. And click *Play* button.

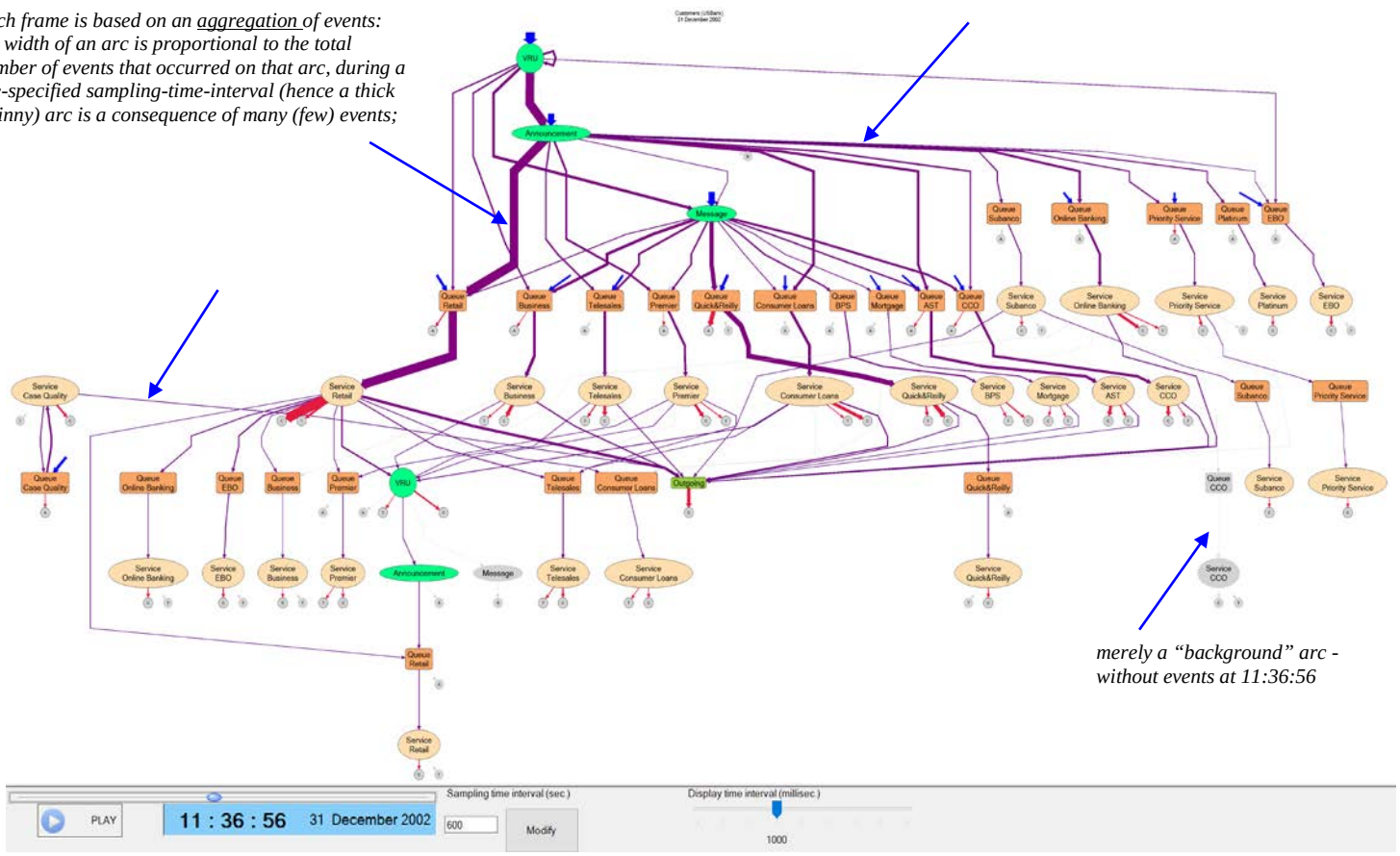


2.3 The Network (Structure) view (USBank Call Center)

Data sample - December 31, 2002 *(SEEGraph file: root folder\cust_subcalls_trn_l2_network.txt)*
To see the graph clearly, simply double click on the picture, which will open the figure in pdf format.

To see the graph clearly, click here

Each frame is based on an aggregation of events: the width of an arc is proportional to the total number of events that occurred on that arc, during a pre-specified sampling-time-interval (hence a thick (skinny) arc is a consequence of many (few) events;



merely a "background" arc - without events at 11:36:56

The above (sub)view captures the graph-structure of a service process, and it does so both statically and dynamically. *Statically* (which is the first "picture" shown when the above sub-view is activated), one observes connections (arcs) among nodes, with thickness of arcs related to their traffic-rates. *Dynamically*, one observes how these traffic-rates change over the day, with the user determining the frequency by which data is sampled by 600 seconds (10 minutes); or displayed - consequent updates of the picture every 1000 milliseconds (1 second).

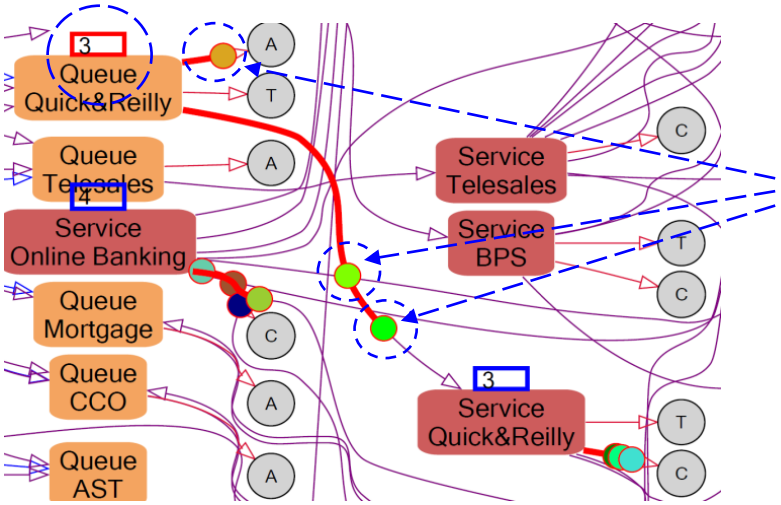
2.4 The Hybrid (Process-Mining + Operations-Research) view

2.4.1 Customers-Calls Flow (USBank Call Center)

Data sample - December 31, 2002 (SEEGraph file: root folder \cust_subcalls_trn_l2_hybrid.txt)

Examples of counting customers in-queue and in-service:

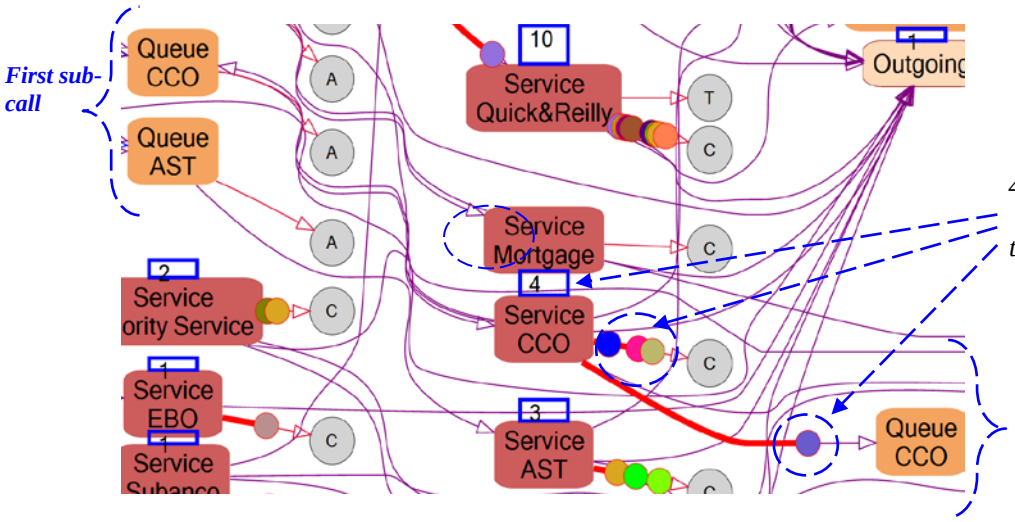
1. ***Counting in Queue***



3 customer calls in Quick&Reilly queue:
1 abandons, 2 waiting before being served.

"Red" *indicates that at least one of the 3 customers waited over T=30 seconds. (Here T is a design-parameter of the graph.)*

2. ***Counting in Service***

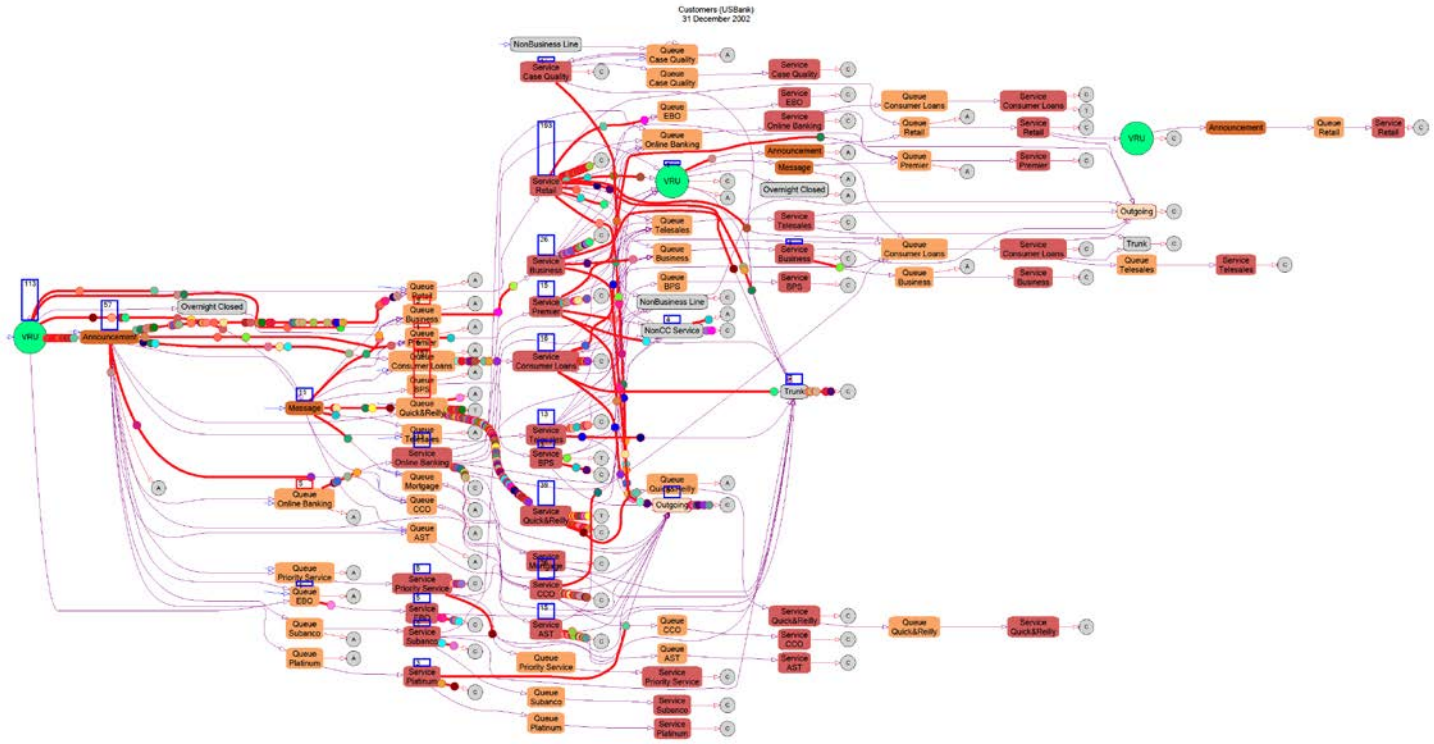


4 customer calls in CCO service:
3 will complete service; 1 will return to the CCO queue (second sub-call).

Second sub-call

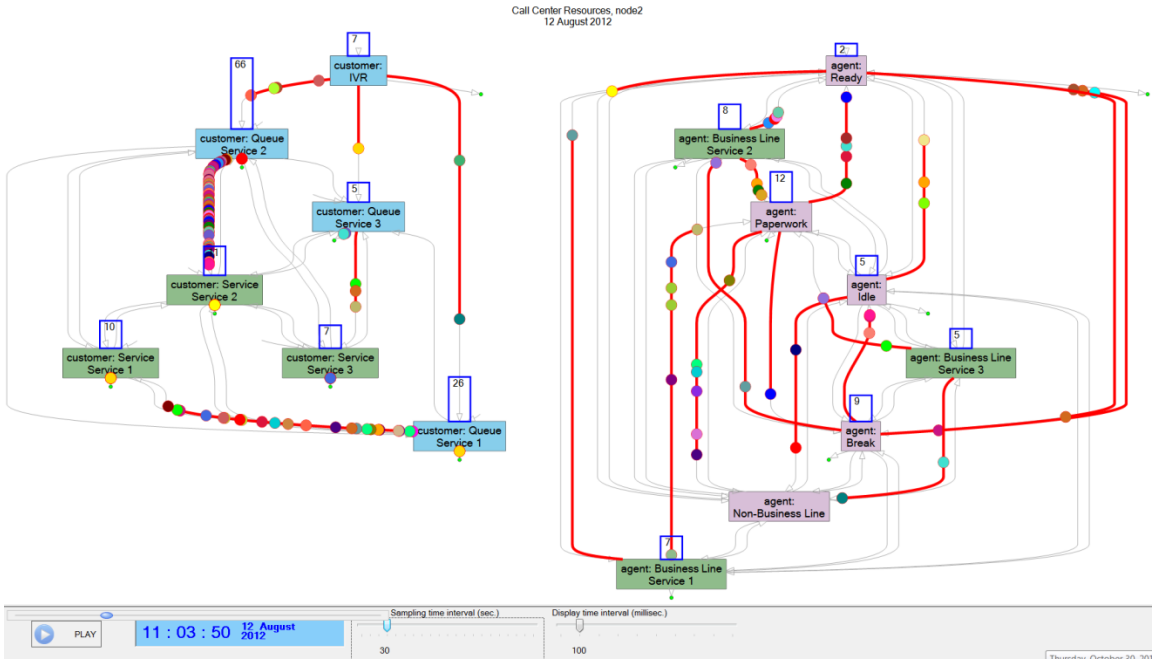
Note: One can also represent a return-to-queue by an arc that feeds back; since there are only a few such returns, it makes sense to acknowledge each of them individually and in sequence.

To see the graph clearly, click here



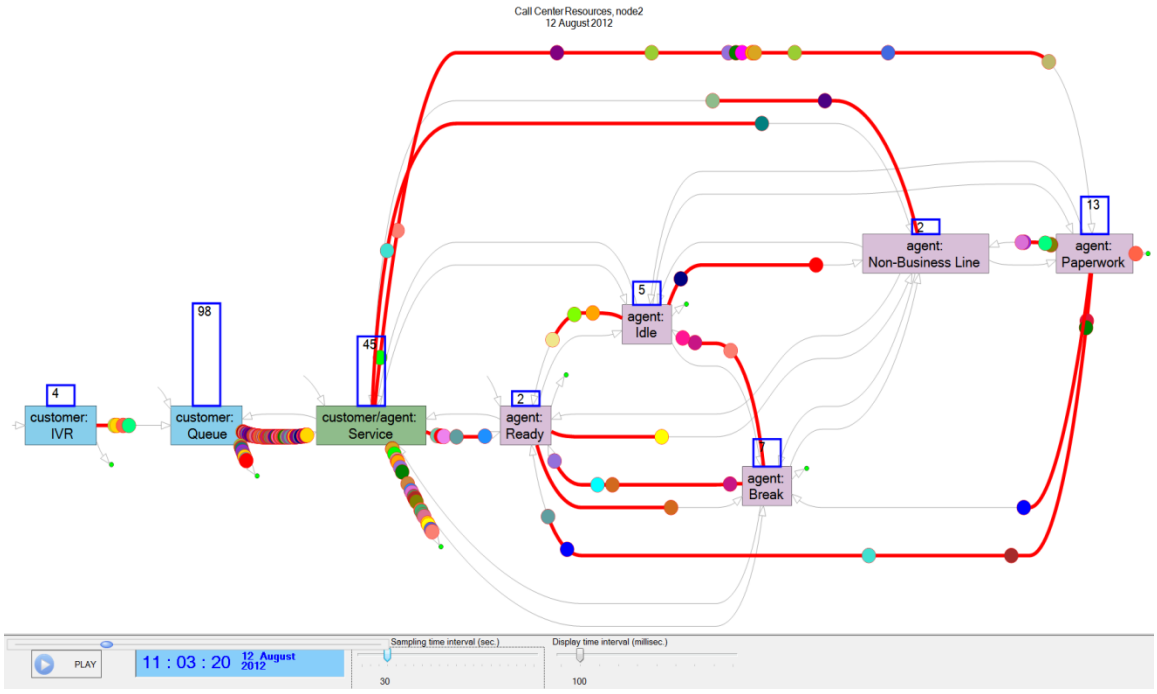
2.4.3 Call Center as a Resource-Network (ILDUBank Call Center)

Customers and Servers, each with their own service network / SEEnimation
SEEGraph file: root folder\Resources_ILDU_hybrid_node2_v1.txt



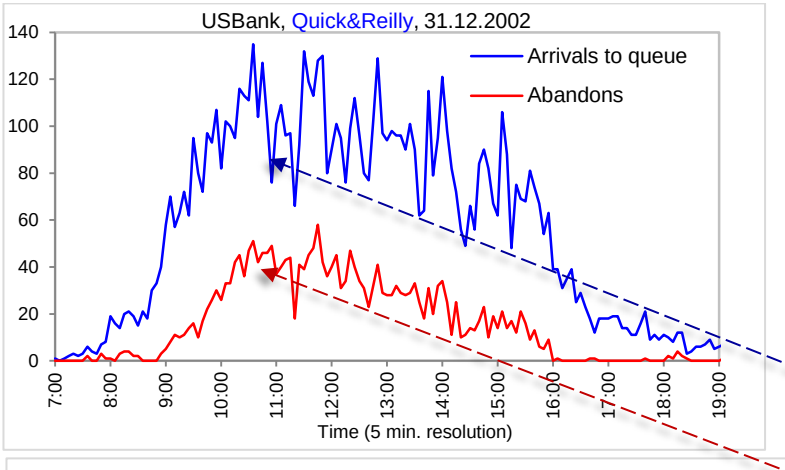
Customers and Servers combined (Symmetrically modelled and empirically (not theoretically) explored in **Erlang-S: A Data-Based Model of Servers in Queueing Networks**, by Azriel, M. and Feigin, accepted to Mgt Science

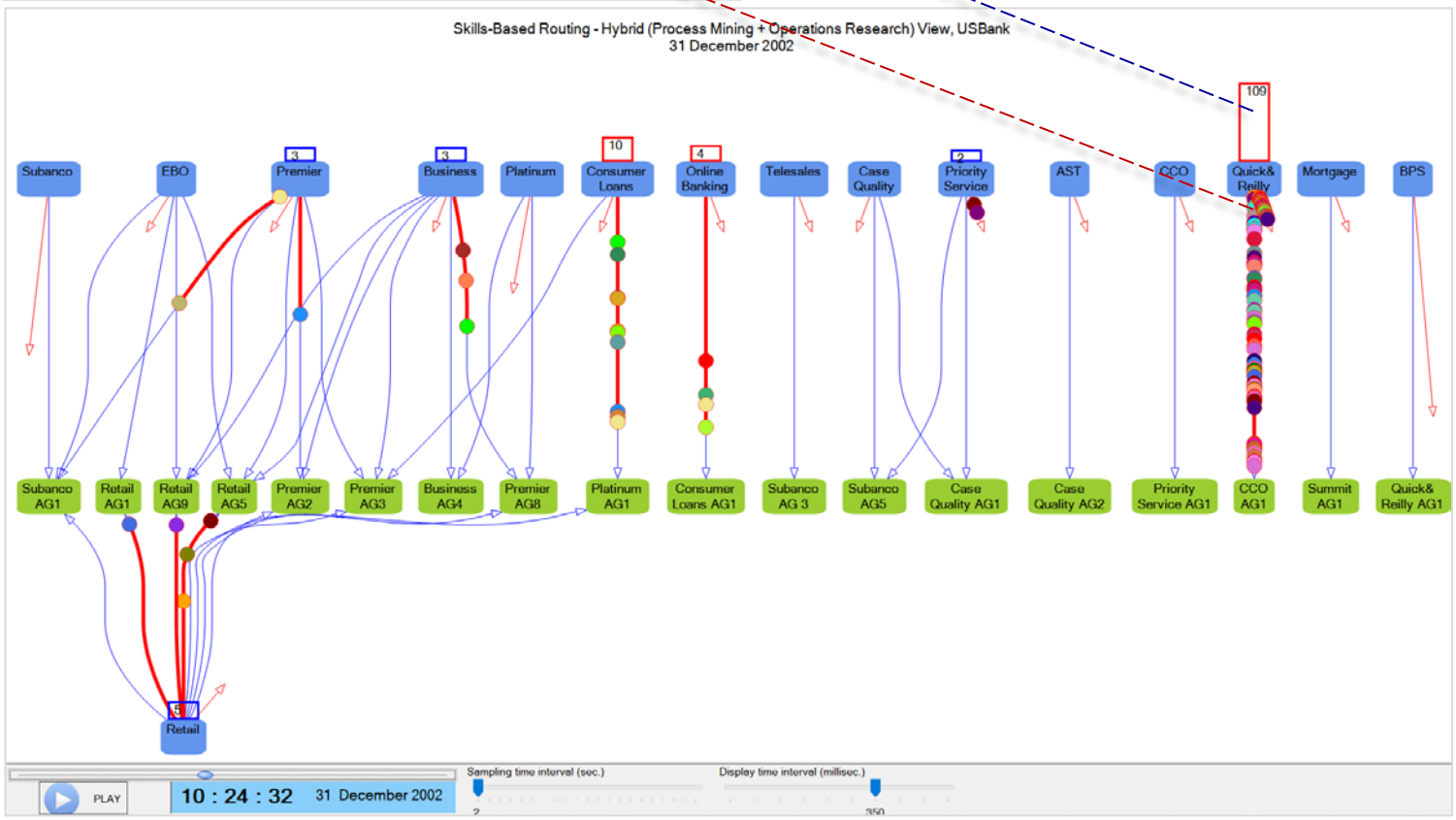
SEEGraph file: root folder\Resources_ILDU_hybrid_node2_v2.txt



2.4.4 Bipartite Graph: Skills-Based Routing (SBR) (USBank Call Center)

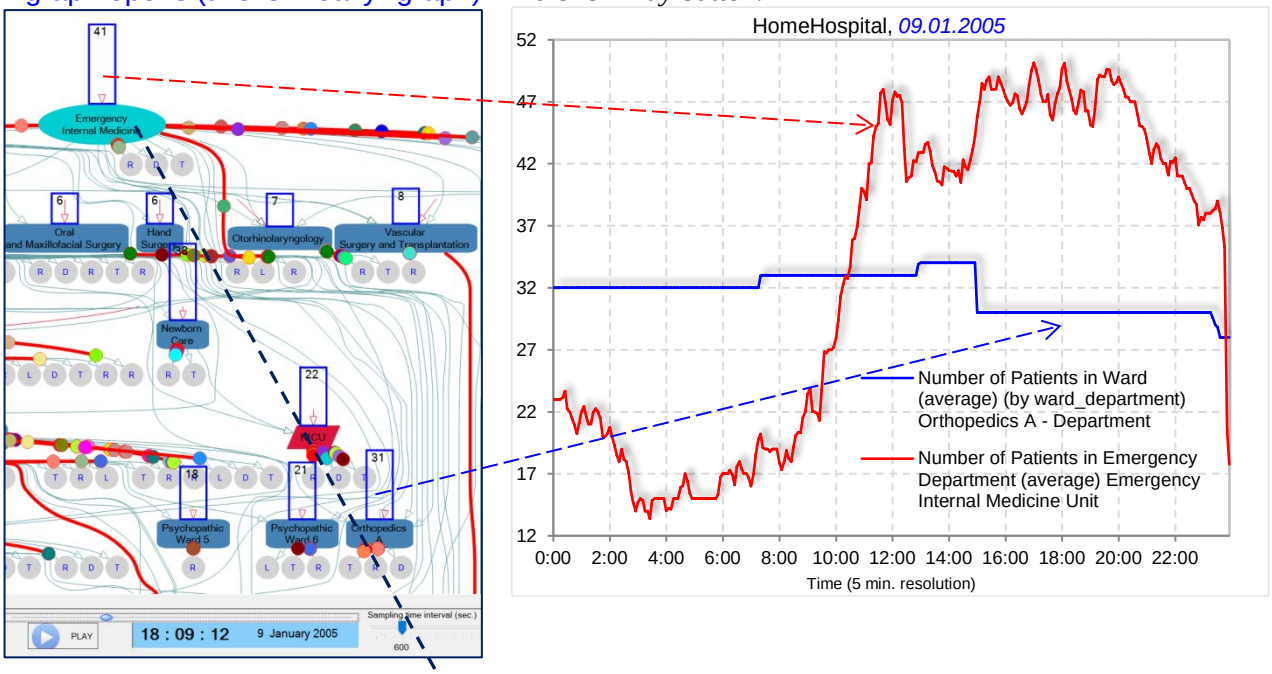
Data sample - December 31, 2002 (SEEGraph file: root folder \ SBR_sinksH.txt)





2.4.5 Patients Flow / Patient Counts in ED/Wards (HomeHospital)

Data sample – all records of patients who were in hospital sometime during January 2005
(SEEGraph file: root folderH \Patients_HomeHospital_hybrid.txt) Select *Layout*. Please wait few minutes until the graph opens (this is "heavy" graph). And click *Play* button.

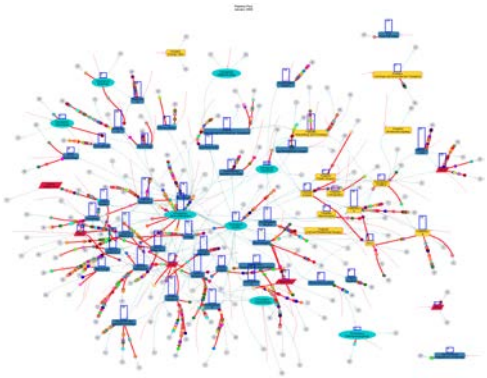


Hierarchical Layout-we can see relationships between departments clearly

To see the graph clearly, click here

SFDP Layout- departments with strong relationships are close by

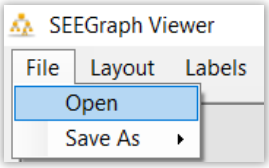
To see the graph clearly, click here



Appendix 1 User Guide

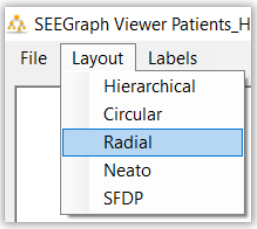
1 Opening a graph/animation file

Click *File -> Open* and choose the desired animation from the available animation list.



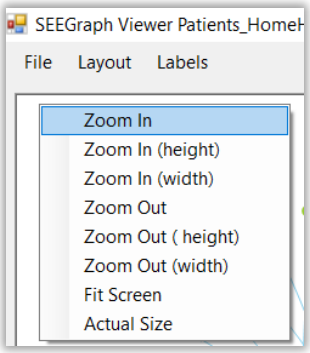
Press the *Play* button on the bottom left side of the screen to run the animations.

2 Select layout



3 Zooming

Right clicking the screen will open the following dialog box.

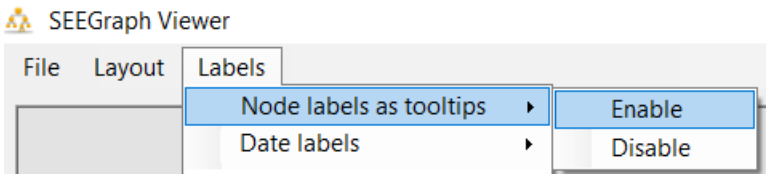


Choose the desired option (e.g. Zoom-in, Zoom-in height) and right-click as needed.

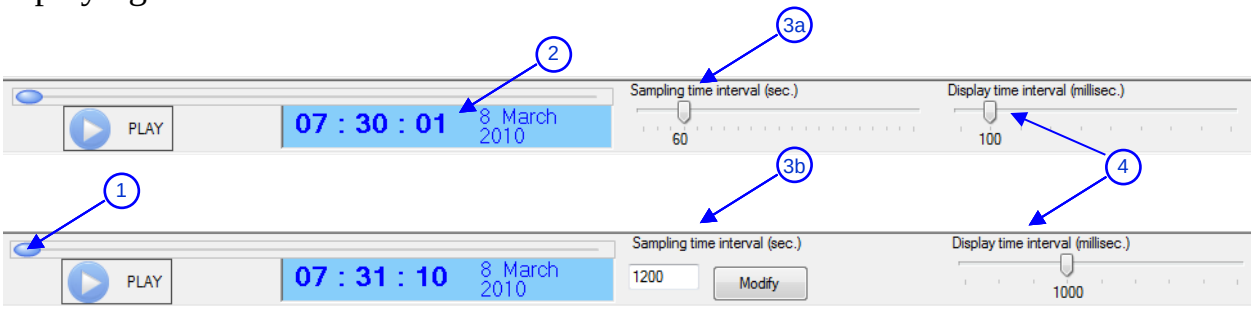
Fit Screen takes you back to the original size.

4 Labels

To see a node-label clearly, select *Labels -> Node labels as tooltips -> Enable*; click on nodes.



5 Displaying Tools



(1) Time scroll

(2) Current time in hh:mm:ss dd/month/yyyy format.

(3) Sampling time interval (sec.) – for example: 60 seconds means that if current time is 7:31:10 then the next picture will be from time 7:32:10. Time intervals are defined during designing of a graph. For example base interval for hospital data can be 600 seconds (10 minutes); base interval for call centers data can be 10 seconds. Following the sampling rules:

   a. Sampling time interval for *Process-Mining (flow), OR (queueing)* and *Hybrid views*: here each frame represents the system-state (snapshot) at the end of a pre-specified sampling-time-interval.

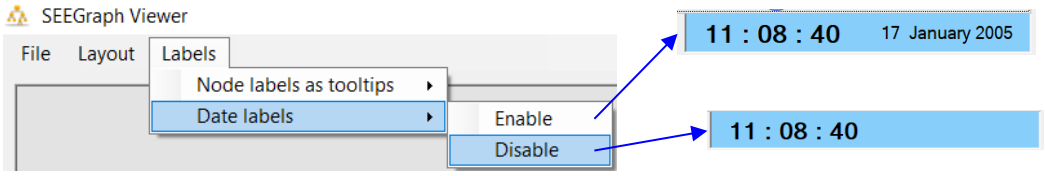
   b. Sampling time interval for *Network View*: SEEnimation that display the Network view (pure structure) have file names that include the word Network. In these SEEnimations, each frame is based on an aggregation of events: the width of an arc is proportional to the total number of events that occurred on that arc, during a pre-specified sampling-time-interval (hence a thick (skinny) arc is a consequence of many (few) events; no-events is represented by merely a "background" arc).

   In all 4 views, the length of the sampling-time-interval can be modified (even during an active SEEnimation) via adjustment of the 'Sampling time interval' bar: specifically, one types in a value in Network, and scrolls the bar in the other 3 cases.

(4) Display time interval (milisec.) – time between consequent updates of the picture (1000 millsec. = 1 second). This adjustment bar controls the time between displays of consequent animation frames. Increased (decreased) 'Display time interval' value makes the animation run slower (faster).

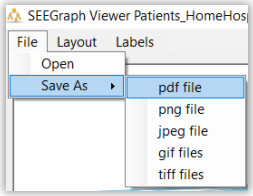
6 Display or DO-NOT display DATE

In order to not display the date (day, month, and year), select *Labels -> Date labels-> Disable*. (This option has been used to adhere to constraints by our SEEParners, whose SEEnimations were used in research papers or pubic lectures.)



7 Save graph

To print animations first run animation (press *Play* button-> *Pause* button) and then *File-> Save As* (pdf format is recommended high quality file).



Appendix 2 Creating Graphs and Networks in SEEGraph – an Toy-Example

We now create a toy-graph from the hospital records of 6 patients.

1. Preparing Data, which is input to graph designer (SEEGraph), and which constitutes 90% of the total work.

   One starts with the definitions of paths and nodes:
   - definition of paths: by patient_id
   - definition of nodes: by department
   - order by patient_id and record_id (left); or alternatively by patient_id and entry_date (right). The former suffices for (static) graph while the latter is required for SEEnimations.

patient_id	record_id	department
1	1	Emergency Internal Medicine Unit
1	2	Internal Medicine A
2	1	Emergency Internal Medicine Unit
2	2	Internal Medicine A
3	1	Emergency Internal Medicine Unit
3	2	Internal Medicine A
4	1	Emergency Internal Medicine Unit
4	2	Internal Medicine E
4	3	Department of Cardiology
4	4	Internal Medicine E
5	1	Emergency Internal Medicine Unit
5	2	Internal Medicine C
5	3	Intensive Cardiac Care Unit
5	4	Department of Cardiology
6	1	Department of Rheumatology

OR

patient_id	entry_date	department
1	02/01/2005 16:38	Emergency Internal Medicine Unit
1	02/01/2005 23:55	Internal Medicine A
2	04/01/2005 14:07	Emergency Internal Medicine Unit
2	04/01/2005 20:23	Internal Medicine A
3	01/01/2005 15:58	Emergency Internal Medicine Unit
3	01/01/2005 17:42	Internal Medicine A
4	06/01/2005 09:35	Emergency Internal Medicine Unit
4	06/01/2005 16:13	Internal Medicine E
4	10/01/2005 08:00	Department of Cardiology
4	11/01/2005 08:00	Internal Medicine E
5	02/01/2005 14:56	Emergency Internal Medicine Unit
5	02/01/2005 21:19	Internal Medicine C
5	07/01/2005 16:37	Intensive Cardiac Care Unit
5	11/01/2005 13:31	Department of Cardiology
6	30/01/2005 08:44	Department of Rheumatology

2. Designing Graph (via SEEGraph software), which starts with a template that includes graph-attributes and connections to the supporting data. Attributes include graph type (static or animation), time interval/horizon (for animation), title, background color, display on full- or partial-screen; *nodes and edges*: counts of nodes and edges, node labels, fill-color for nodes and edges, line color and style, shape, size, font style and color; *clusters* (if relevant): fill color, line color and style, font style and color, …

The Template, with its supporting Data, enables the creation of the following SEEGraph output (saved in a text file):

```
START GRAPH
DIRECTED 1
RANKDIR 0
NODSEP 72
RANKSEP 8
EXACT_RANKSEP 0
LABEL sample"for"tutorial"(HomeHospital)\nJanuary"2005
FILLED 1
FILLCOLOR 137
PENWIDTH 0
PENCOLOR 0
LINESTYLE 0
FONTNAME 1
FONTSIZE 8
FONTSTYLE 0
FONTCOLOR 0
ANIMTYPE 0
EDGEPENWIDTH 1
EDGEFILL 1
START NODES
NNODES 12
ATTR height
0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5  0.5
ATTR width
0.75  0.75  0.75  0.75  0.75  0.75  0.75  0.75  0.75  0.75  0.75  0.75
ATTR label
Entry  Emergency\nInternal"Medicine"Unit  Internal\nMedicine"A  Internal\nMedicine"C  Internal\nMedicine"E  Department\nof"Cardiology  Intensive\nCardiac"Care"Unit
Department\nof"Rheumatology  "  "  "  "
ATTR shape
1  1  1  1  1  1  1  3  3  3  3
ATTR fillcolor
77  124  124  124  124  124  124  124  53  53  53  53
ATTR fill
1  1  1  1  1  1  1  1  1  1  1  1
ATTR pencolor
8  8  8  8  8  8  8  8  8  8  8  8
ATTR penstyle
0  0  0  0  0  0  0  0  0  0  0  0
ATTR penwidth
1  1  1  1  1  1  1  1  1  1  1  1
ATTR fontname
1  1  1  1  1  1  1  1  0  0  0  0
ATTR fontsize
18  18  18  18  18  18  18  18  0  0  0  0
ATTR fontstyle
0  0  0  0  0  0  0  0  0  0  0  0
ATTR fontcolor
8  8  8  8  8  8  8  8  0  0  0  0
END NODES
START EDGES
NEDGES 13
0  0  1  1  4  5  1  3  6  2  4  5  7
1  7  2  4  5  4  3  6  5  8  9  10  11
ATTR label
5  1  3  1  1  1  1  1  1  3  1  1  1
ATTR arrowhead
3  3  3  3  3  3  3  3  3  3  3  3  3
ATTR fill
0  0  0  0  0  0  0  0  0  0  0  0  0
ATTR fillcolor
0  0  0  0  0  0  0  0  0  0  0  0  0
ATTR penwidth
2  2  2  2  2  2  2  2  2  2  2  2  2
ATTR penstyle
0  0  0  0  0  0  0  0  0  0  0  0  0
ATTR pencolor
8  8  8  8  8  8  8  8  8  8  8  8  8
ATTR fontname
1  1  1  1  1  1  1  1  1  1  1  1  1
ATTR fontsize
18  18  18  18  18  18  18  18  18  18  18  18  18
ATTR fontstyle
0  0  0  0  0  0  0  0  0  0  0  0  0
ATTR fontcolor
8  8  8  8  8  8  8  8  8  8  8  8  8
ATTR weight
2.23607  1  1.73205  1  1  1  1  1  1.73205  1  1  1
ATTR minlen
5  5  5  5  5  5  5  5  5  1  1  1  1
END EDGES
END GAPH
```

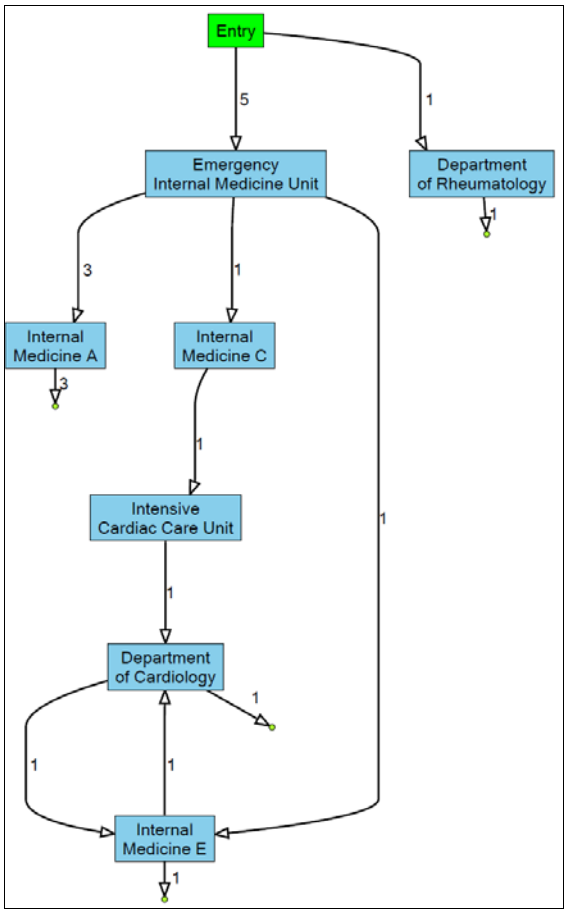
Note that the above starts with a list of graph attributes, then attributes of nodes (there are 12 nodes, each with its individual attributes), and finally of edges (13 of them).Dynamic graphs require one more template of times.

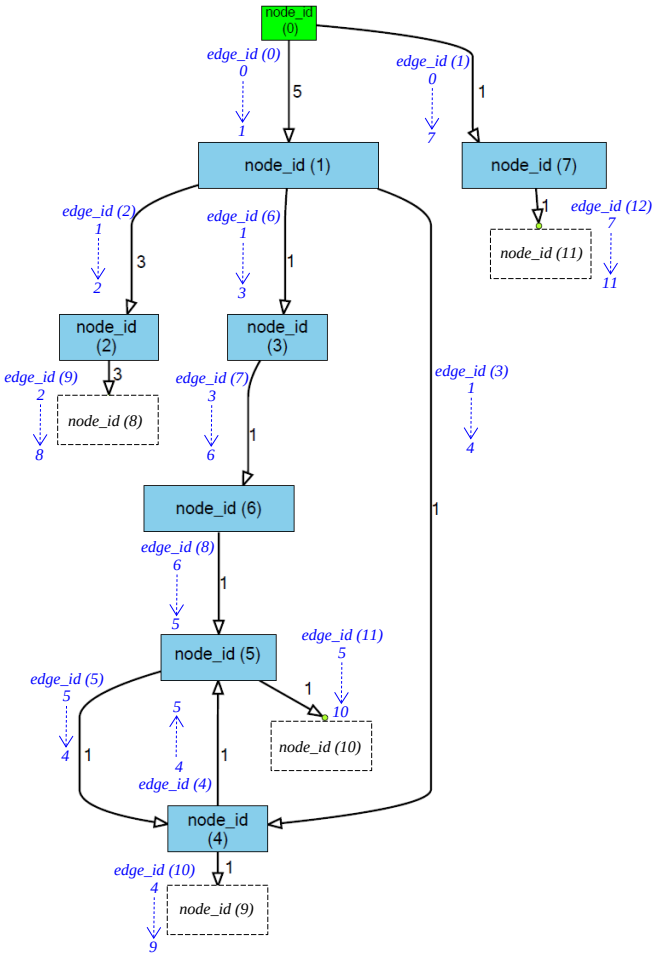
3. Displaying Graph or Network/SEEnimation (via the SEEGraph Viewer software). This requires specification of the layout type (sub-structure – which is one of the 5 layouts, presently available in the Viewer); this layout, jointly with the information created in Step 2 above, enables graph-display, which is supported by a powerful design-engine. For example, this engine optimizes graph-layout ("picture complexity") by seeking to minimize edge-intersections; it is based on technologies that control the motion of AGVs over a factory-floor but ensuring that AGVs do not collide with each other.

Structure, namely nodes and arcs and their relations:

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

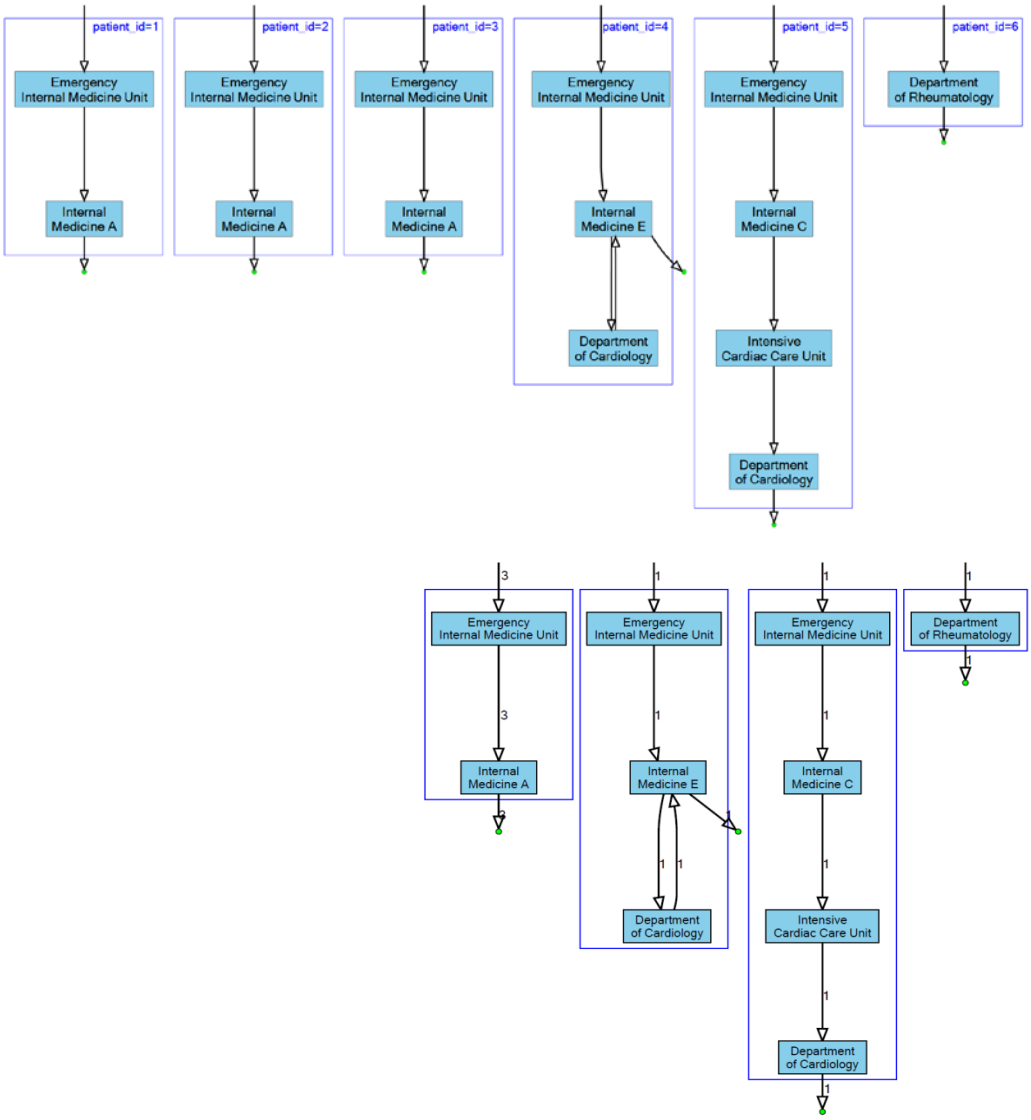
	edge_id (0)	*edge_id (1)*	*edge_id (2)*	*edge_id (3)*	*edge_id (4)*	*edge_id (5)*	*edge_id (6)*	*edge_id (7)*	*edge_id (8)*	*edge_id (9)*	*edge_id (10)*	*edge_id (11)*	*edge_id (12)*
node_id	0	0	1	1	4	5	1	3	6	2	4	5	7
node_id	1	7	2	4	5	4	3	6	5	8	9	10	11





On the left: hierarchical graph of 6 patients flow. On the right: schematic representation of nodes and edges.

We also decided to add TWO more graphs: one that has ALL 6 patients separately; and one that pools together the 3 patients with equal paths (creating patient care-paths).


patient_id=1
Emergency Internal Medicine Unit
Internal Medicine A
patient_id=2
Emergency Internal Medicine Unit
Internal Medicine A
patient_id=3
Emergency Internal Medicine Unit
Internal Medicine A
patient_id=4
Emergency Internal Medicine Unit
Internal Medicine E
Department of Cardiology
patient_id=5
Emergency Internal Medicine Unit
Internal Medicine C
Intensive Cardiac Care Unit
Department of Cardiology
patient_id=6
Department of Rheumatology
3
Emergency Internal Medicine Unit
3
Internal Medicine A
3
1
Emergency Internal Medicine Unit
1
Internal Medicine E
1
1
1
Department of Cardiology
1
Emergency Internal Medicine Unit
1
Internal Medicine C
1
Intensive Cardiac Care Unit
1
Department of Cardiology
1
1
Department of Rheumatology
1