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 Designer and SEEGraph Viewer

Avishai Mandelbaum
Valery Trofimov
Igor Gavako
Arik Senderovich
Ella Nadjharov
Nitzan Yuviler

Created: May 22, 2014
Last Revised: February 1, 2021

Contents

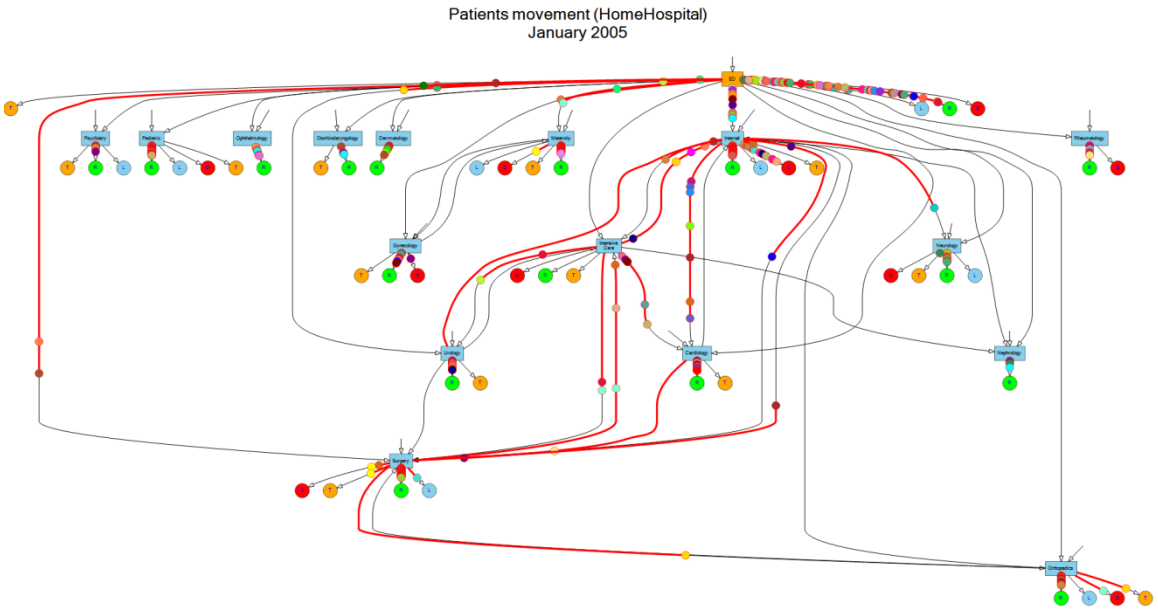
Part I SEEGraph: User Guide

1. Before you start designing graphs, it is necessary to define first your user extension, if it was not defined yet. ***(See*** *Appendix 1, Creation of SEEStat user extension).*

2. This is a beta version of SEEGraph 1.0. Special graph structures are not installed automatically. Please ask the SEELab team to prepare the needed structures.

Design graph: Flow graph type (one node dimension)

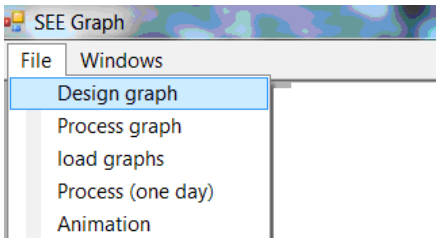
Patients Flow (flow animation), HomeHospital data



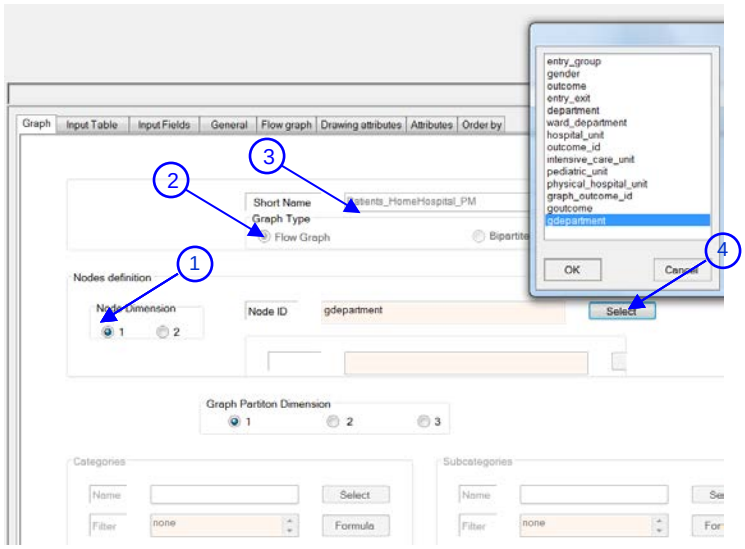
Example of data:

visit_details				
medical_id	department	entry_time	exit_time	outcome
20000310	1	1104833437	1104859349	2
20000310	39	1104859349	1105104780	7
20000312	1	1104782151	1104818427	2
20000312	37	1104818427	1104937200	7
20000321	1	1105004120	1105028039	2
20000321	43	1105028039	1105344000	3
20000321	46	1105344000	1105430400	3
20000321	43	1105430400	1105452720	7

Open SEEGraph 1.0. Select *File->Design graph.* Select the *HomeHospital* study. Click *OK.*

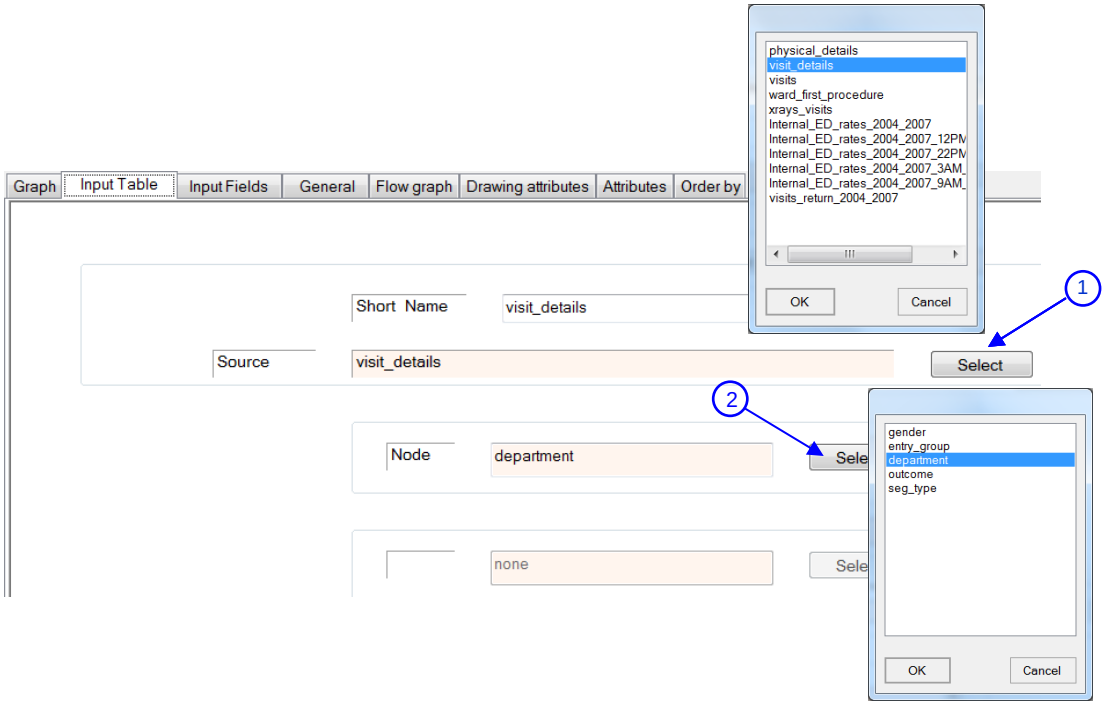


The Graph tab:



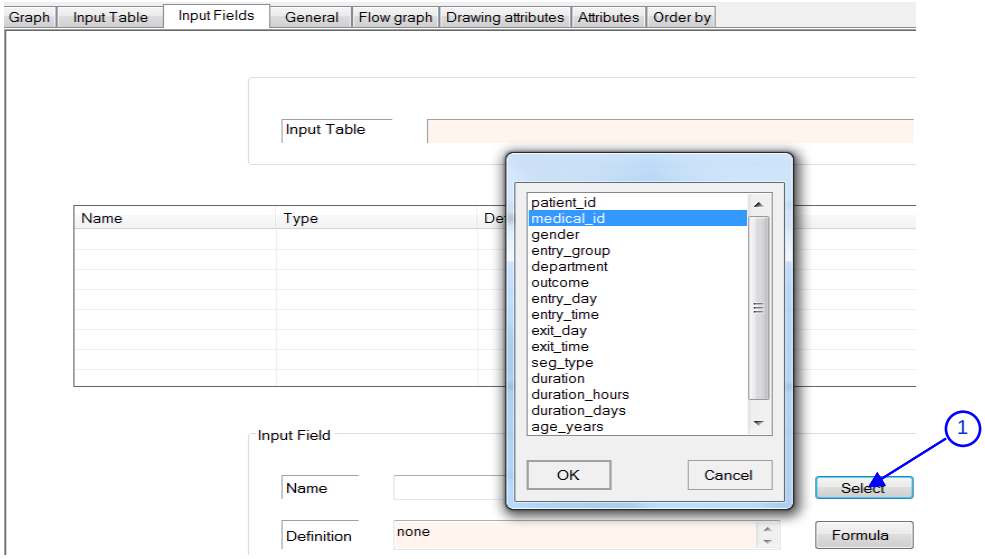
(1) Node Dimension: *1* (default value)
(2) Select graph type: *Flow Graph* (default value)
(3) Fill graph short name: *Patients_HomeHospital_PM*
(4) Select dictionary for Node ID: click on the *Select* button and select *gdepartment* dictionary, click OK.-> OK

The Input Table tab:



(1) Click on the *Select* button (Source), select *database* and then select *visit_details* table, click OK.
(2) Click on the *Select* button (Node) and select *department* field. Click OK.

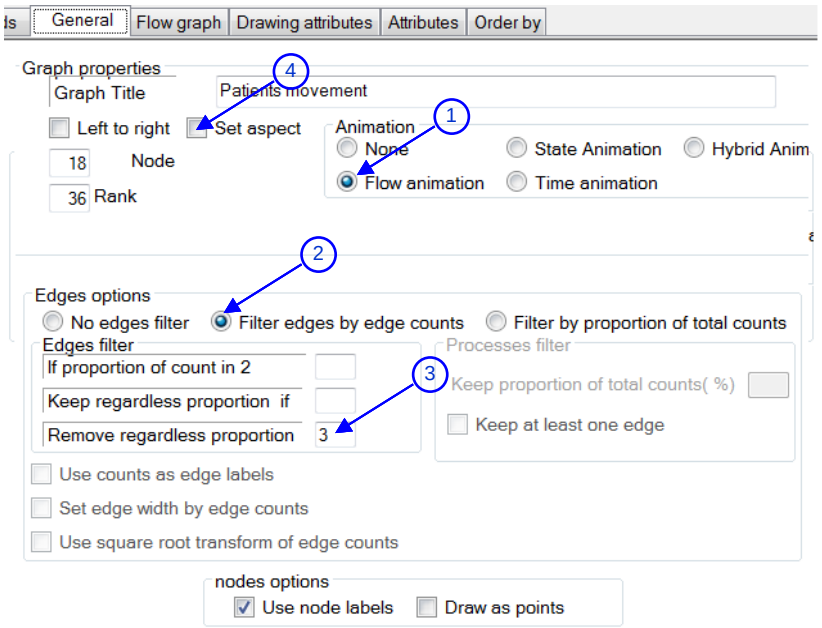
The Input Fields tab:



(1) Click on the *Select* button and select *medical_id* field, click OK. In the same manner, select the following fields: *entry_time, outcome, exit_time.*

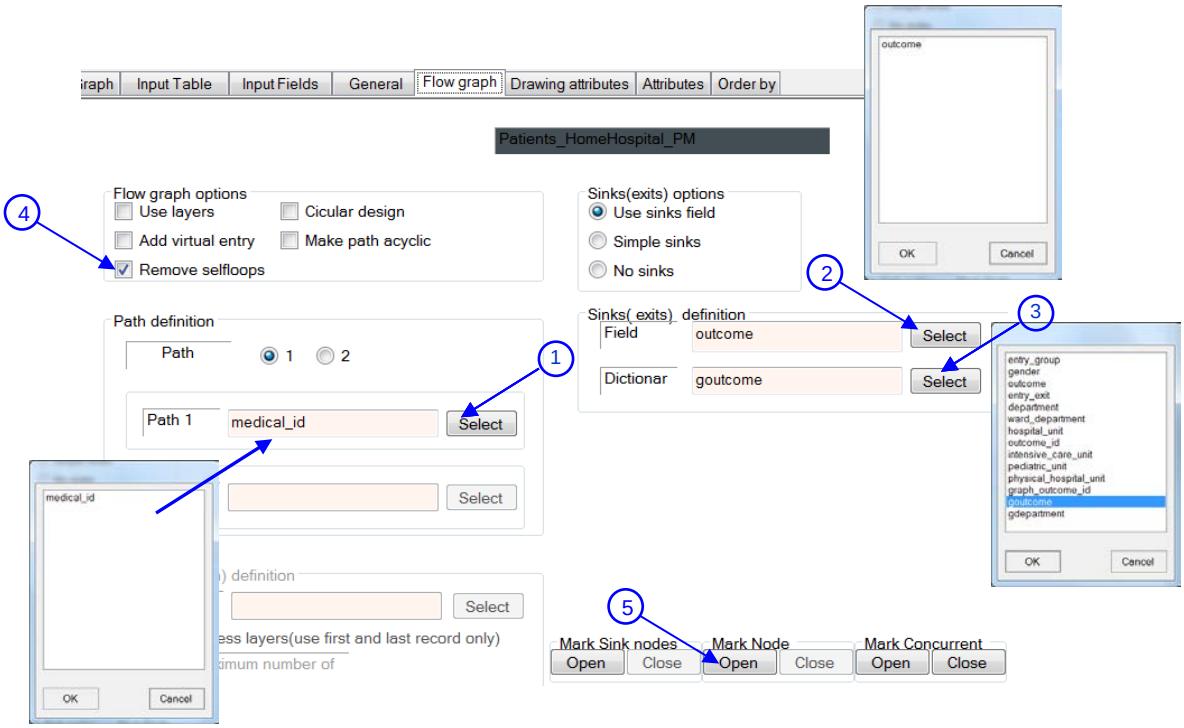
The General tab:

See detailed description of General tab in Appendix 2.

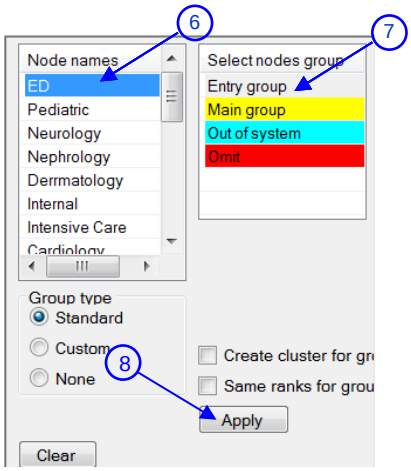


(1) Select animation type: *Flow animation*
(2) Select edges options : *Filter edges by edge count*
(3) Remove regardless proportion if count < *3*
(4) *Set aspect* – full screen resolution

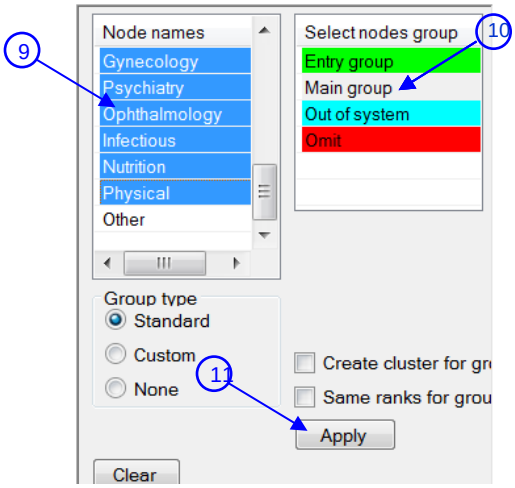
The Flow graph tab:



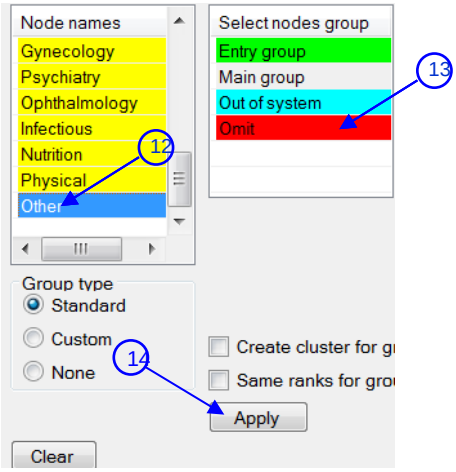
(1) Select path definition by medical_id: click on the *Select* button, select *medical_id* field, and click OK.
(2) Select field for sinks (exits) definition: click on the *Select* button, select *outcome* field, and click OK.
(3) Select dictionary field for sinks (exits) definition: click on the *Select* button, select *goutcome* dictionary, and click OK.
(4) In *Flow graph options* select *Remove self loops.*
(5) In *Mark Node* options click on the *Open* button.



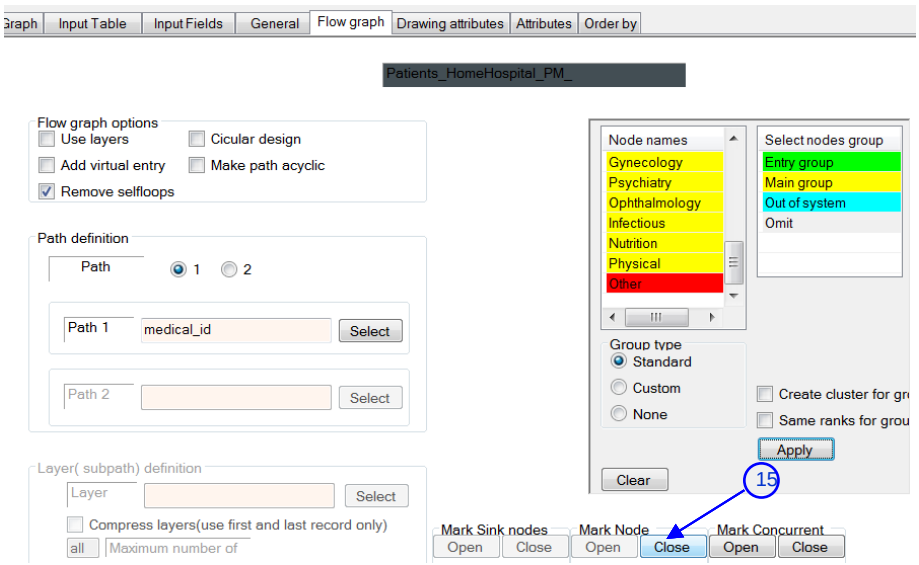
(6) Select *ED* from the list of *Node names* (on the left side)
(7) In *Select nodes group* (on the right side) select *Entry group.*
(8) Click on the *Apply* button



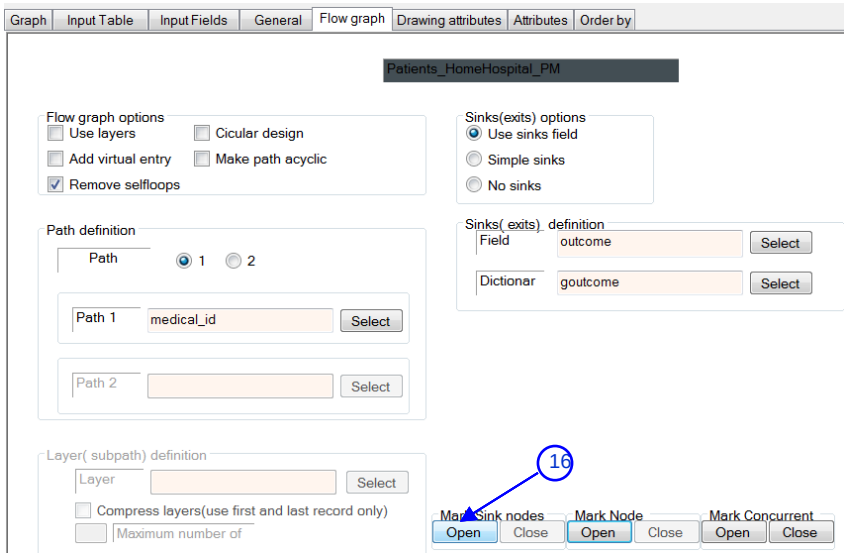
(9) From the list of Node names (on the left side), select all departments except for *ED* and *Other*.
(10) In *Select nodes group* (on right side), select *Main group.*
(11) Click on the *Apply* button



(12) Select *Other* department from the list of *Node names* (on the left side)
(13) In *Select nodes group* (on the right side), select *Omit.*
(14) Click on the *Apply* button.

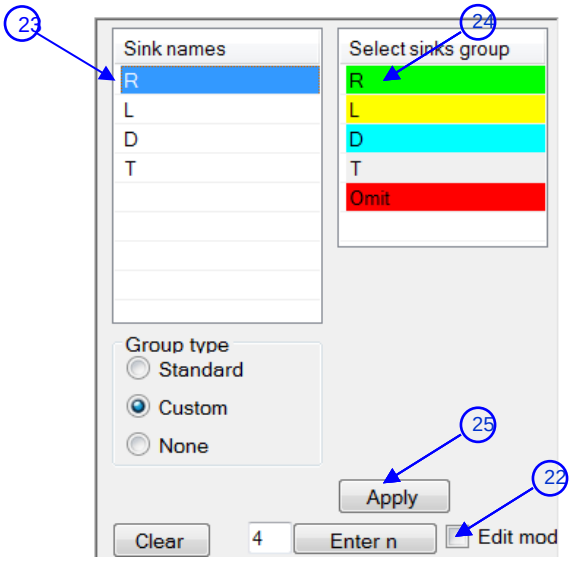


(15) Click on the *Close* button in *Mark Node* options.



(16) Click on the *Open* button in *Mark Sink* nodes options.

(17) Select group type: *Custom*
(18) Write *4* (group)
(19) Click the *Enter n* button.
(20) Select *Edit mode*.
(21) Rename *Group 1* as *R*, in the same manner, rename the other groups: Group 2 as L, Group 3 as D, and Group 4 as T.

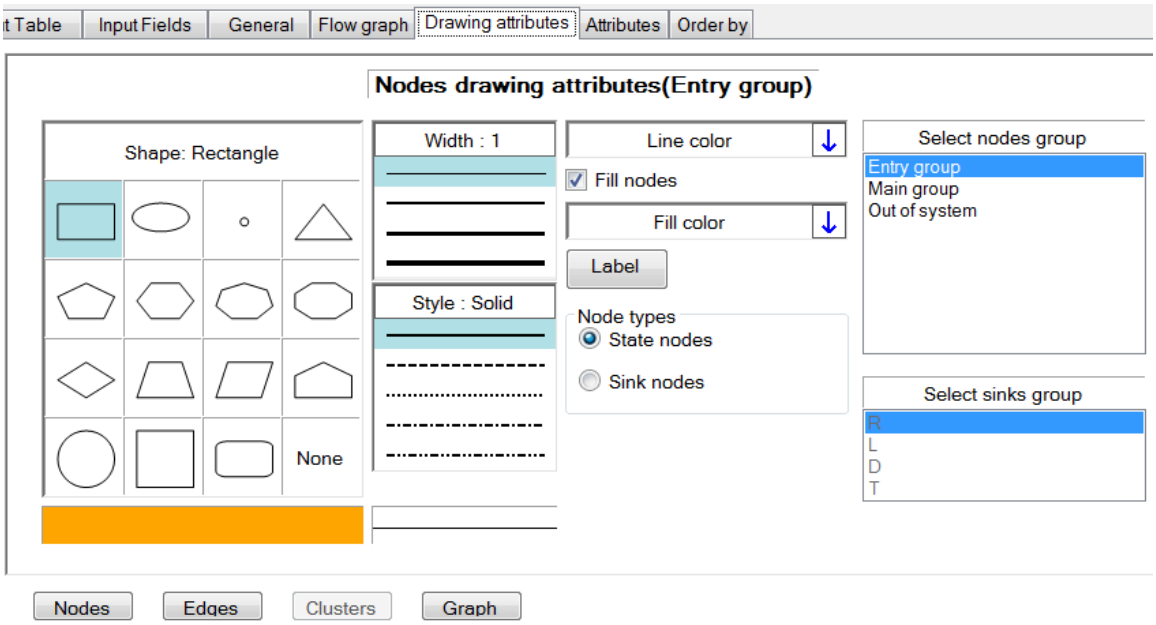


(22) Uncheck *Edit mode.*
(23) In the *Sinks names* list select *R.*

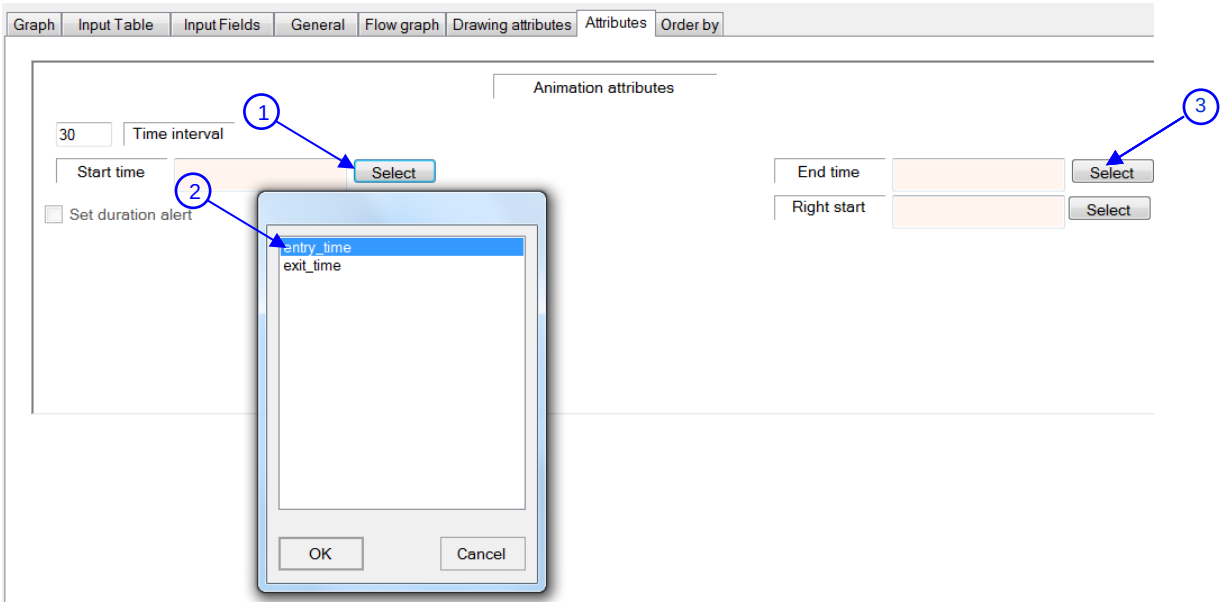
(24) In *Select sinks group* select *R*.
(25) Click on the *Apply* button. Select the other sinks groups in the same manner.

The Drawing attributes tab:

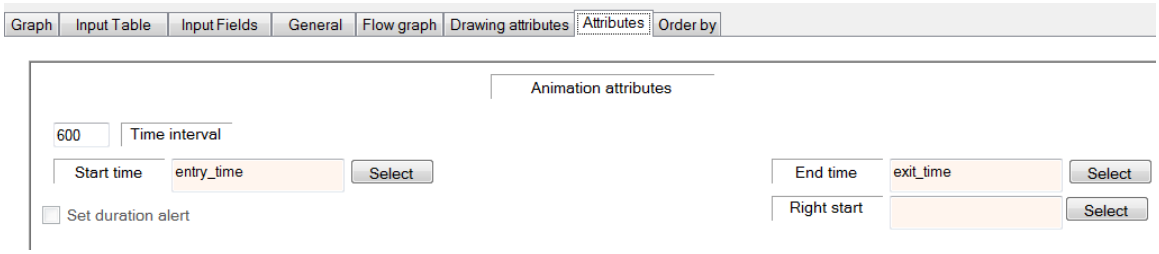
See detailed description of Drawing attributes tab in section: 3 Edit Graph: Drawing attributes tab.



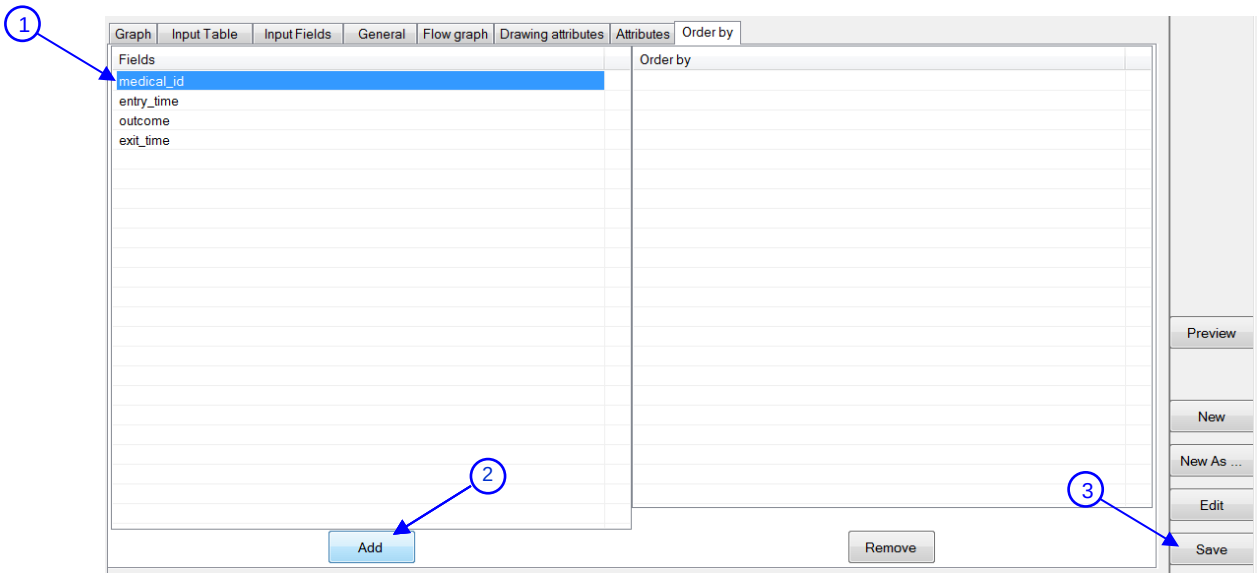
The Attributes tab:



(1) Start time selection: click on the *Select* button (on the left side).
(2) Select *entry_time*, click OK.
(3) End time selection: click on the *Select* button (on the right side) and select *exit_time*, click OK.

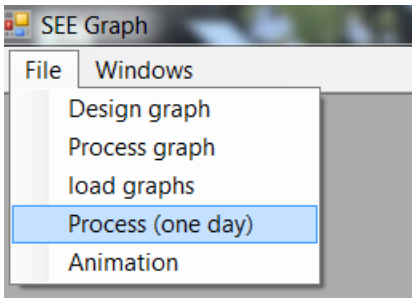


The Order by tab:

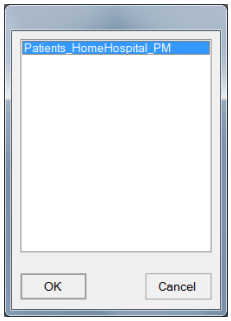


(1) Select *medical_id* field
(2) Click on the *Add* button. Select the *entry_time* field in the same manner.
(3) Click on the *Save* button.

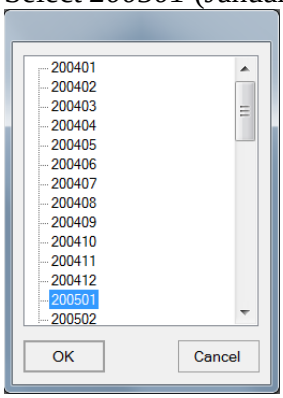
Reopen SEEGraph 1.0.
Select *File->Process (one day).* Select the *HomeHospital* study. Click *OK.*



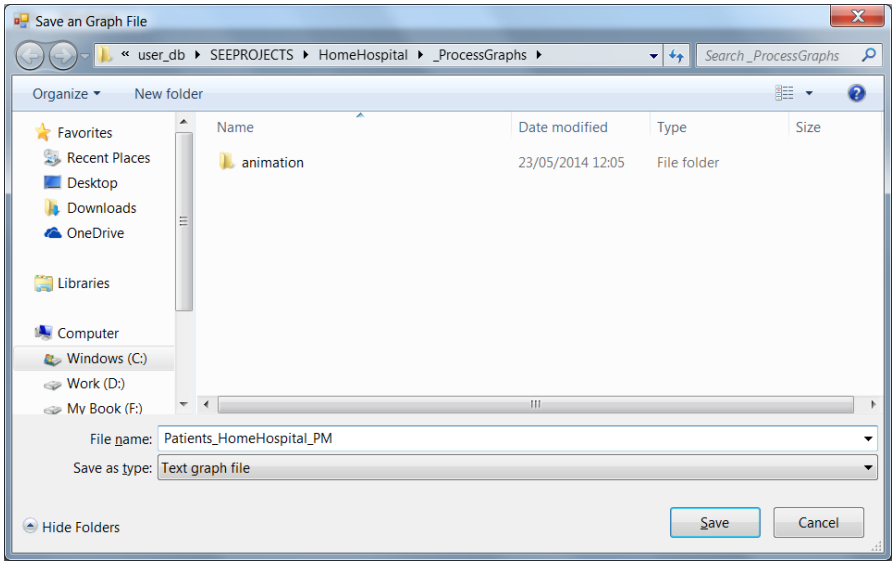
Select *Patients_HomeHospital_PM* graph, click OK.



Select *200501* (January 2005), click OK.



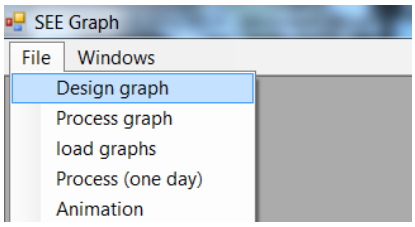
Save file in C:\ animation. Fill the text file name and click Save.



Open *SEEGraph View 1.0.*
Select *File->Open*. Select *Patients_HomeHospital_PM.txt* file from *C:\animation.* Click on the *Open* button.

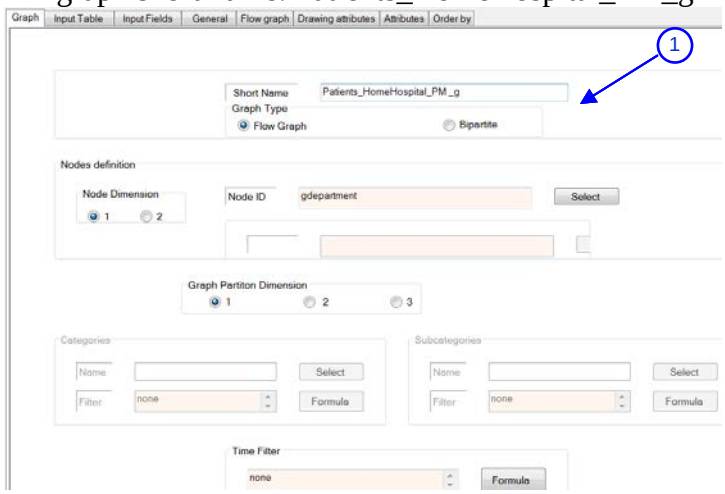
Design Graph: New AS option

Open SEEGraph 1.0. Select *File->Design graph.* Select the *HomeHospital* study. Click *OK.*

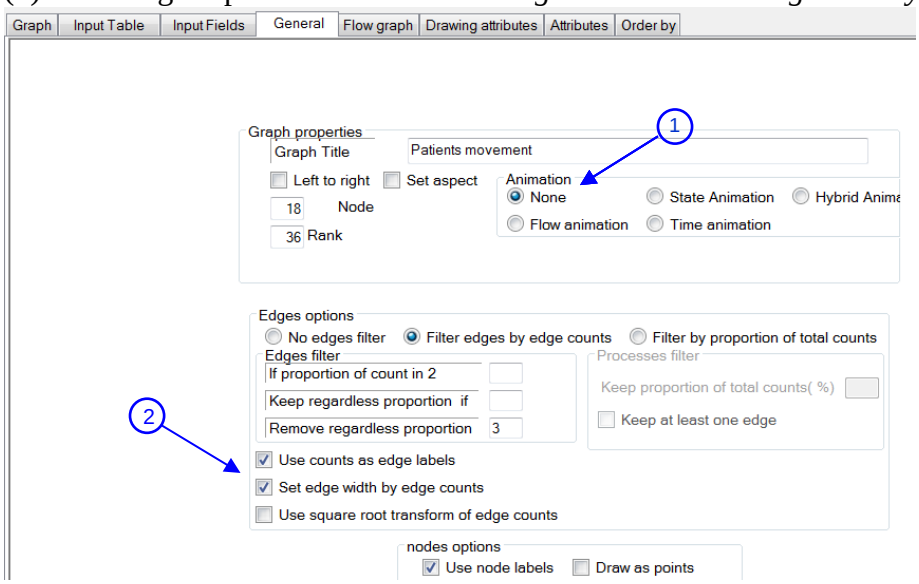


Click on the *New As* button. Select *Patients_HomeHospital_PM* graph. Click OK.

Fill graph short name: Patients_HomeHospital_PM_g in *Graph* tab.



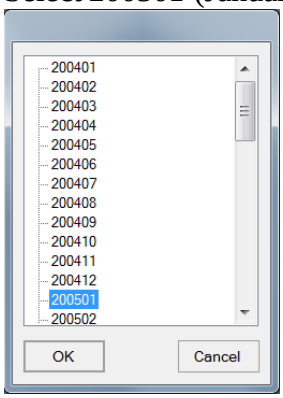
In the *General* tab: (1) Select without animation option: *None*
(2)Select edges options: *Use counts as edge labels* and *Set edge with by edge counts*.



Click *Save* button.

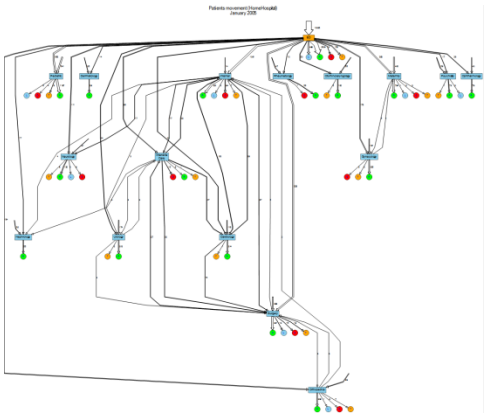
Reopen SEEGraph 1.0.
Select *File->Process (one day).* Select the *HomeHospital* study. Click *OK.*
Select *Patients_HomeHospital_PM_g* graph, click OK.

Select *200501* (January 2005), click OK.



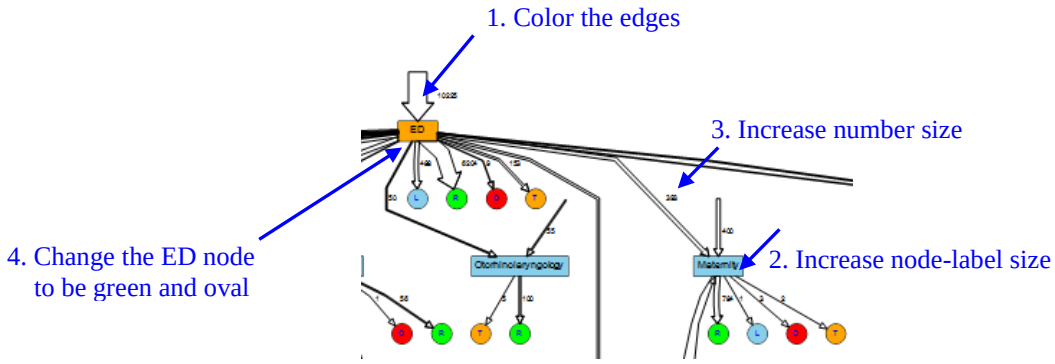
Save file in C:\ animation. Fill the text file name and click Save.

Open *SEEGraph View 1.0.*
Select *File->Open*. Select *Patients_HomeHospital_PM_g.txt* file from *C:\ animation.*
Click on the *Open* button.



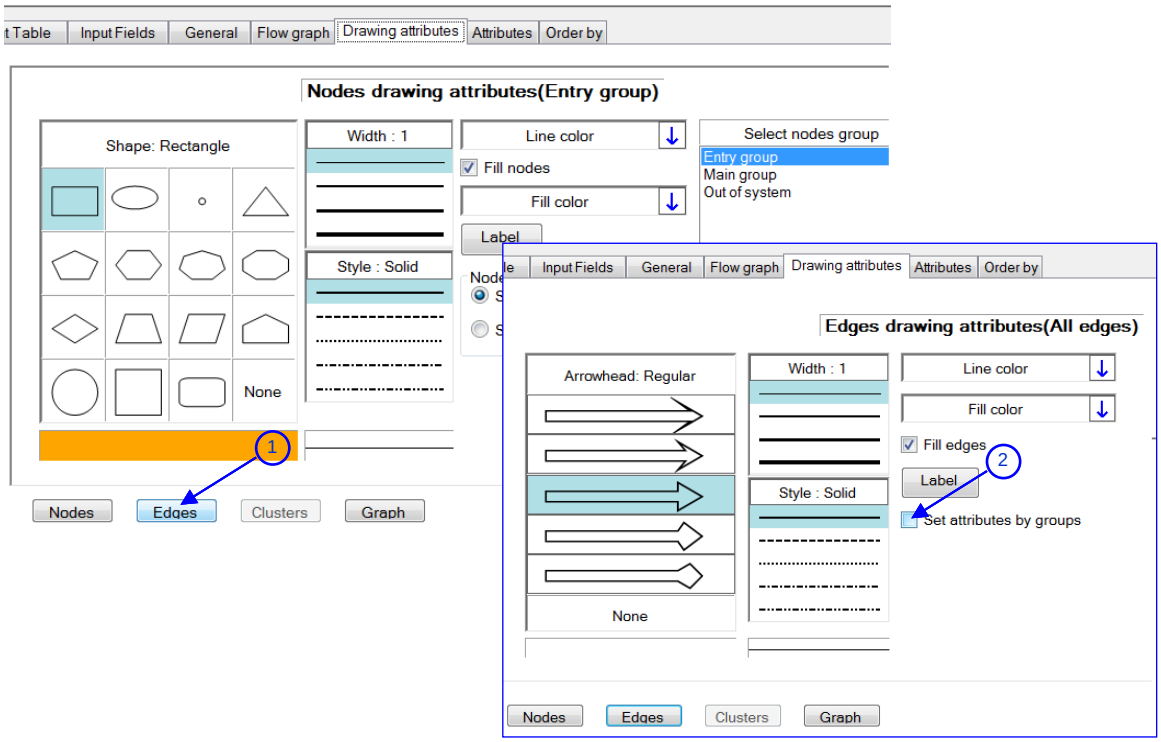
Edit Graph: Drawing attributes tab

Edit tasks:

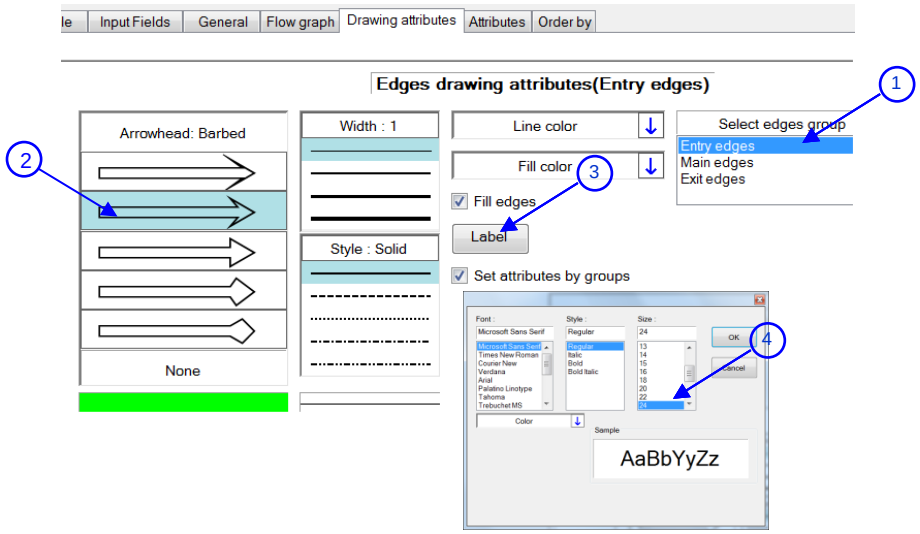


Open SEEGraph 1.0. Select *File->Design graph.* Select the *HomeHospital* study. Click *OK.*
Click on the *Edit* button. Select *Patients_HomeHospital_PM_g* graph. Click OK.

Open the *Drawing attributes* tab.



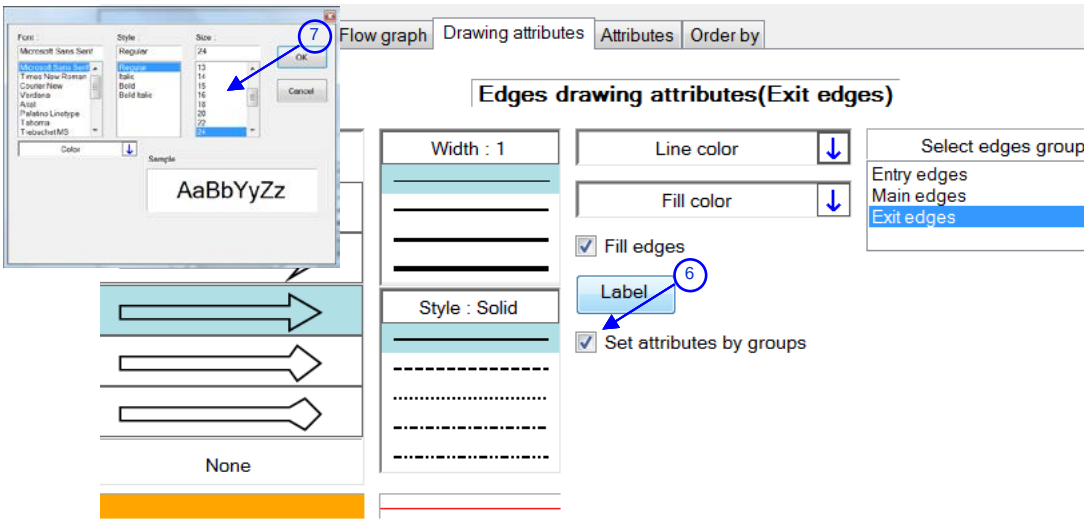
(1) Click on the *Edges* button.
(2) Select *Set attributes by groups.*



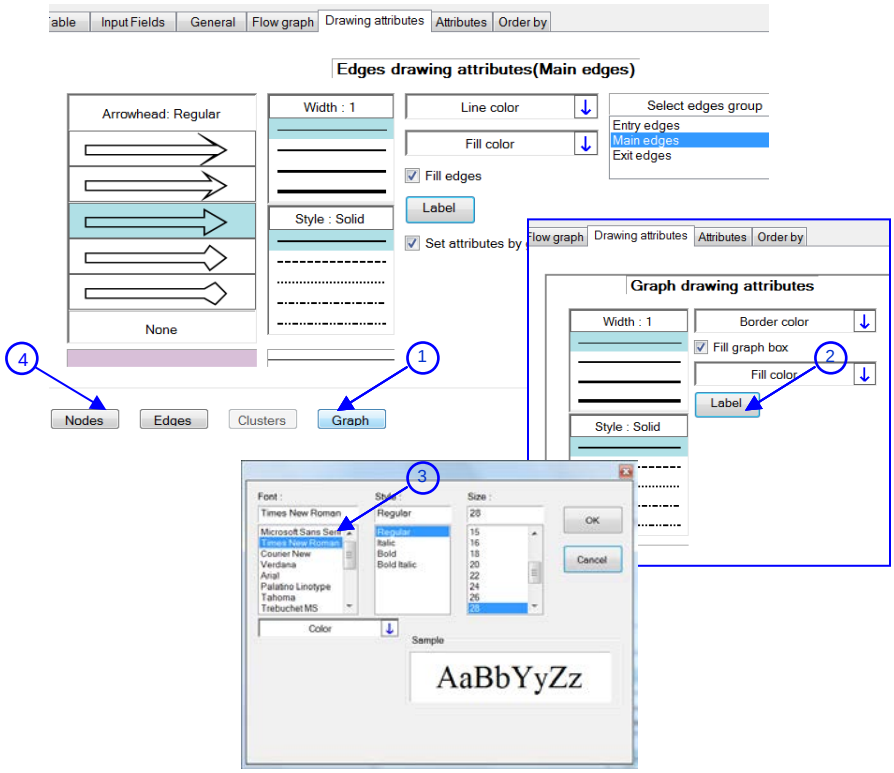
(1) Select edge group: *Entry edges*
(2) Select arrowhead*: Barbed*
(3) Click on the *Label* button
(4) Select font size: *24.* Click OK. Change the font size for the *Main edges* group in the same manner.

(1) Select edge group: *Exit edges*
(2) Click on the arrow in *Fill color*.

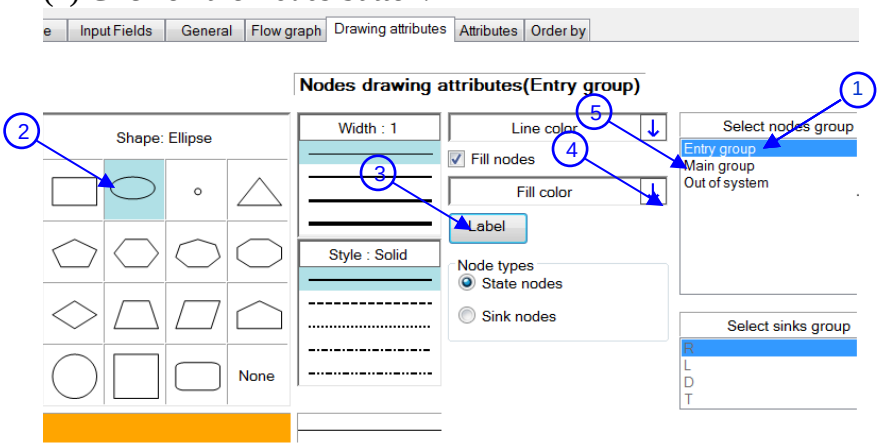
(3) Click on the orange color.
(4) Click on the arrow in *Line color*
(5) Click on the red color.



(6) Click on the *Label* button
(7) Select font size: *24.* Click OK.



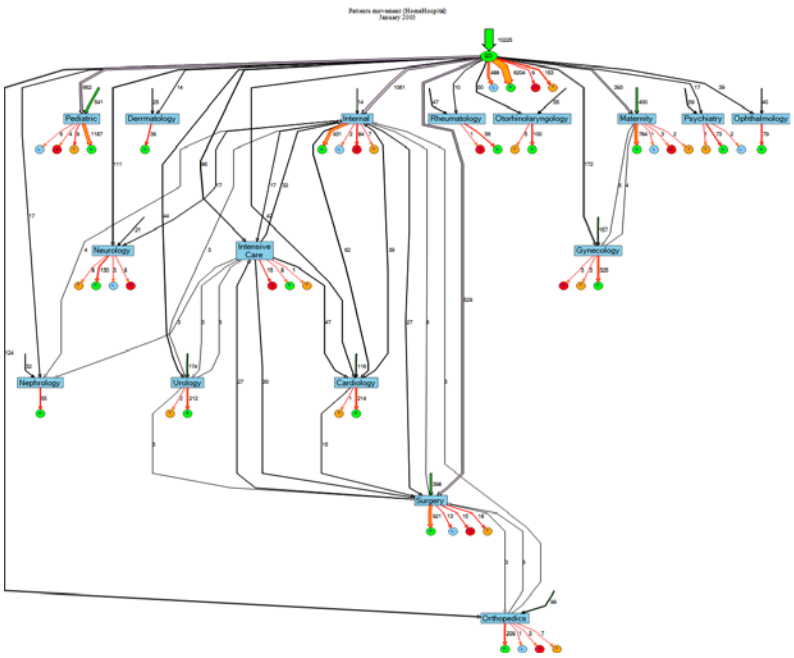
(1) Click on the *Graph* button
(2) Click on the *Label* button
(3) Select *Times New Roman* font. Click OK.
(4) Click on the *Nodes* button.



(1) Select nodes group: *Entry group.*
(2) Select node shape: *Ellipse.*
(3) Click on the *Label* button and select font size 24. Click OK.
(4) Click on the arrow in *Fill color*, and select the lime color.
(5) Select node group: *Main group.* Change font size to 34.

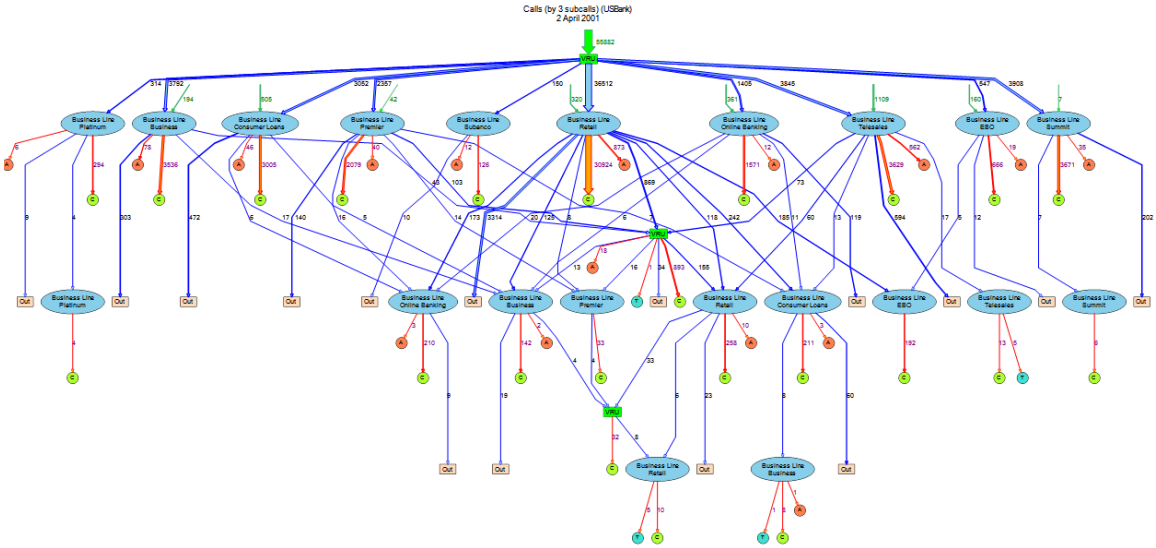
Save graph.
Reopen SEEGraph 1.0 and process graph.
Save file in C:\ animation. Name the text file *Patients_HomeHospital_PM_g_edited.txt* and click Save.

Open *SEEGraph View 1.0.*
Select *File->Open*. Select *Patients_HomeHospital_PM_g_edited.txt* file from *C:\ animation.* Click on the *Open* button.



Design graph: Flow graph type (two node dimension)

Calls Flow (USBank data)

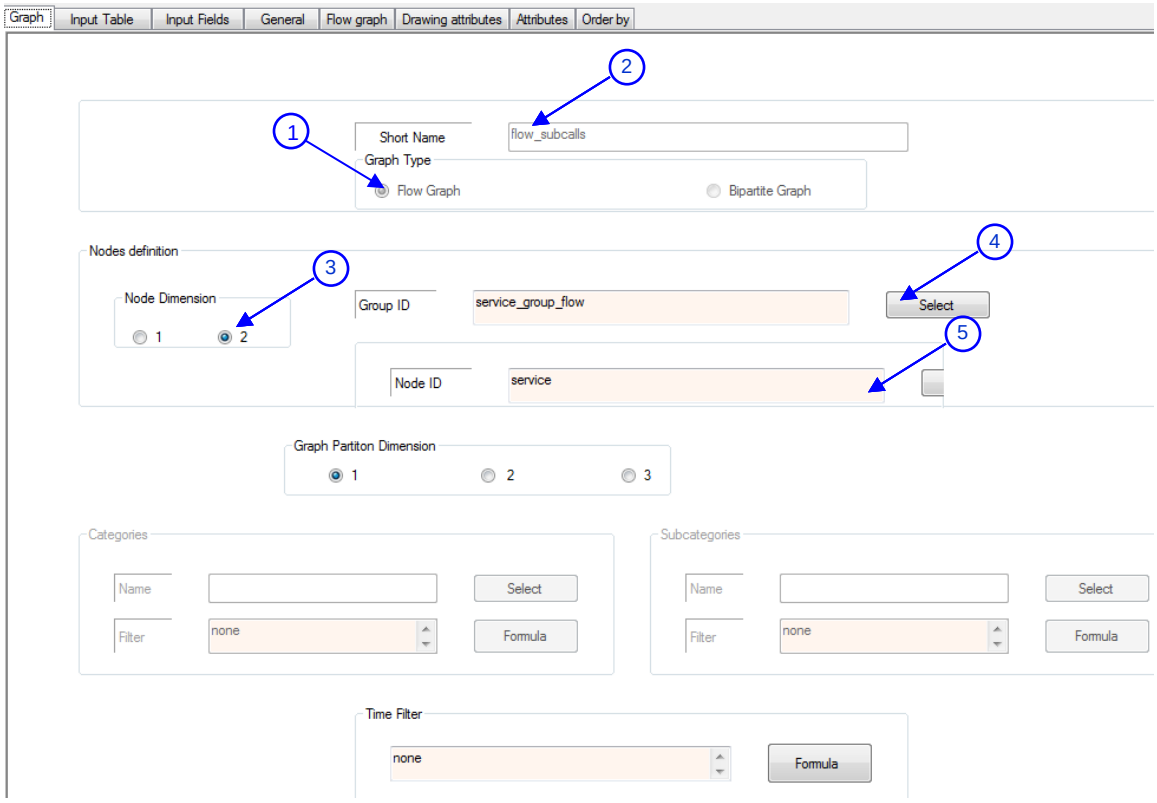


Example of data:

cust_subcalls table					
call_id	segment_start	cust_subcall	service	service_group	outcome
457260248	986255571	1	1	1	22
457260248	986255641	1	1	3	20
457260248	986255671	1	1	2	22
457260248	986255872	2	1	1	22

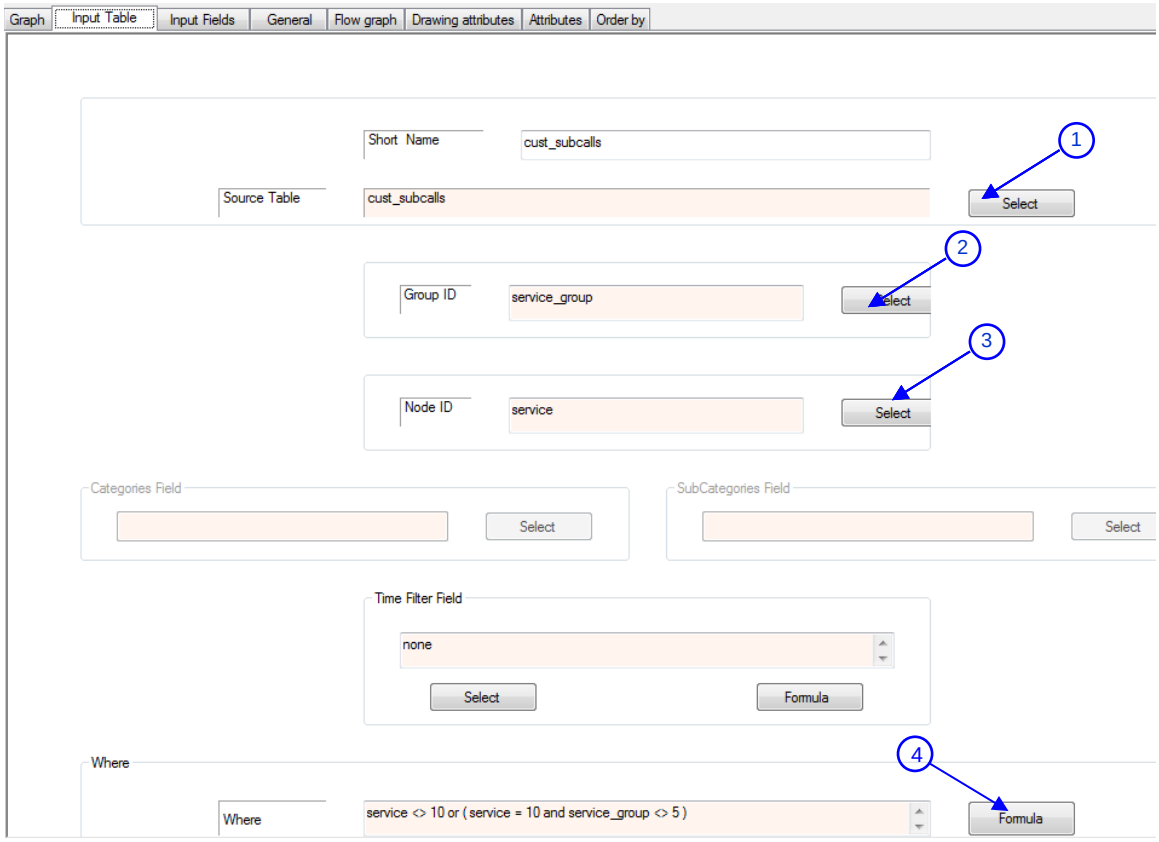
cust_subcalls table					
call_id	segment_start	cust_subcall	service	service_group	outcome
457260248	986256031	2	1	3	20
457260248	986256099	2	1	2	22
457260248	986256373	3	1	1	22
457260248	986256778	3	1	3	20
457260248	986256845	3	1	2	1
457260259	986255591	1	1	1	22
457260259	986255710	1	1	3	20
457260259	986255768	1	1	2	2
457260263	986255593	1	8	3	20
457260263	986255640	1	8	2	22
457260263	986256046	2	8	2	1
457260264	986255595	1	1	1	22
457260264	986255732	1	1	3	20
457260264	986255768	1	1	2	1

The Graph tab:



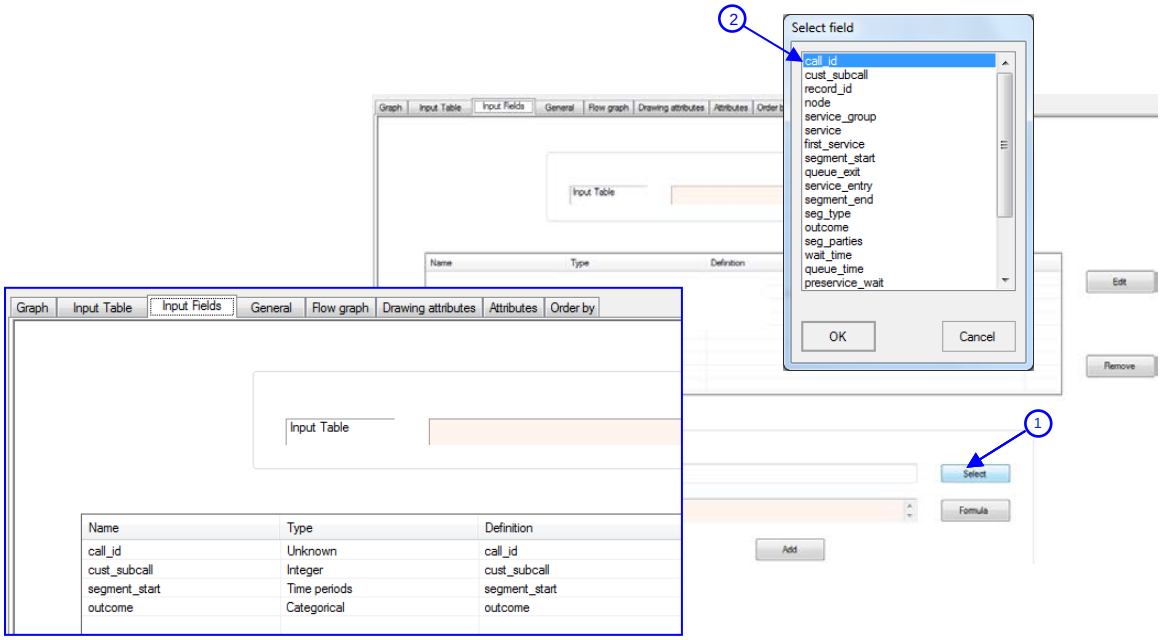
(1) Select graph type: *Flow Graph* (default value)
(2) Fill graph short name: *flow_subcalls*
(3) Select node dimension *2*.
(4) Select dictionary for Group ID: click on the *Select* button and select *service_group_flow* dictionary, click OK.-> OK
(5) Select dictionary for Node ID: click on the *Select* button and select *service* dictionary, click OK.-> OK

The Input Table tab:



(1) Click on the *Select* button (Source), select *database* and then select *cust_subcalls* table, click OK.
(2) Click on the *Select* button (Group ID) and select *service_group* field. Click OK.
(3) Click on the *Select* button (Node ID) and select *service* field. Click OK.
(4) Click on the *Formula* button and fill formula: *service <> 10 or (service = 10 and service_group <> 5)*(see Appendix 5 How to type formulas in SEEStat)

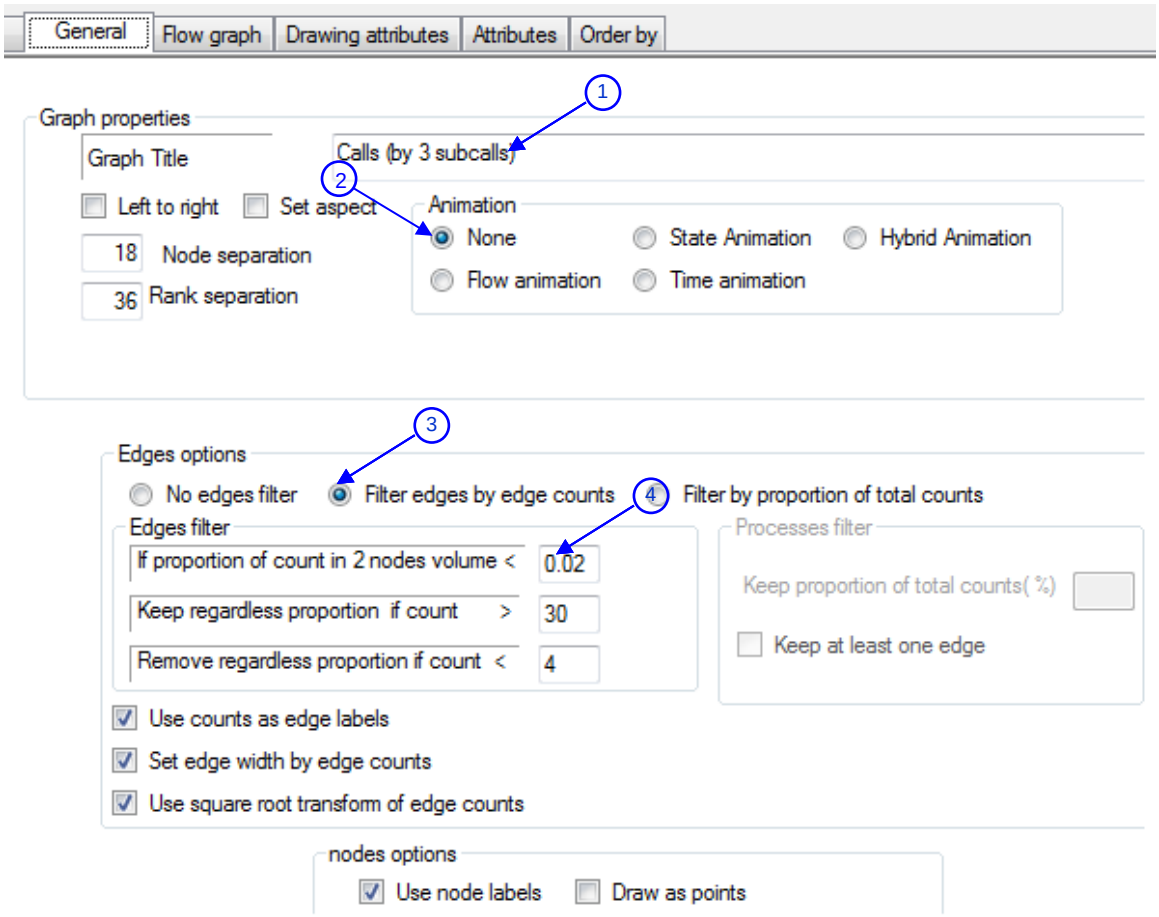
The Input Fields tab:



(1) Click on the *Select* button.
(2) Select *call_id* field, click OK. In the same manner, select the fields: *cust_subcall*, *segment_start*, *outcome.*

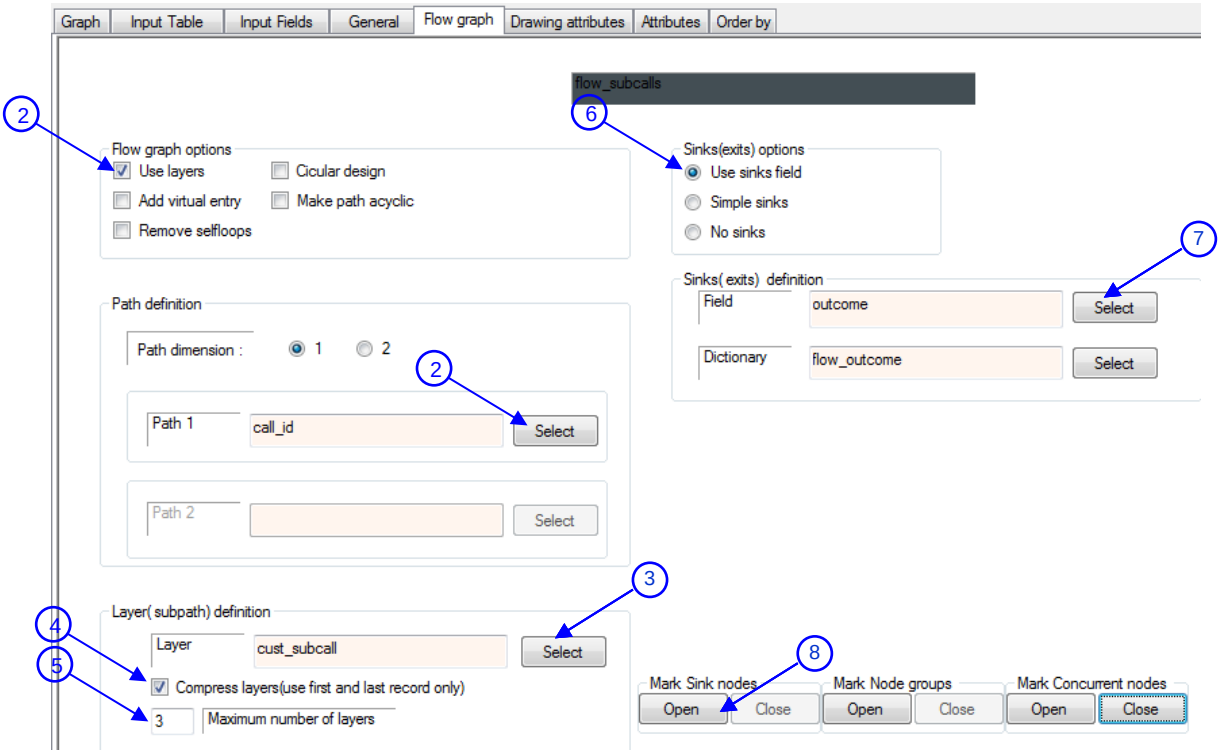
The General tab:

See detailed description of General tab in Appendix 2.

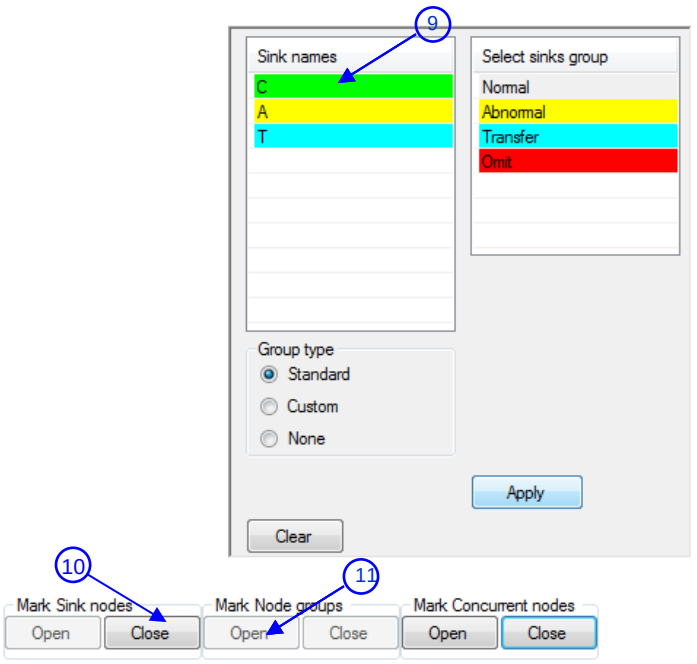


(1) Fill graph title: *Calls (by 3 subcalls)*
(2) Graph without animation: select *None*
(3) Select edges options : *Filter edges by edge count*
(4) Fill edges filters as presented in the picture.

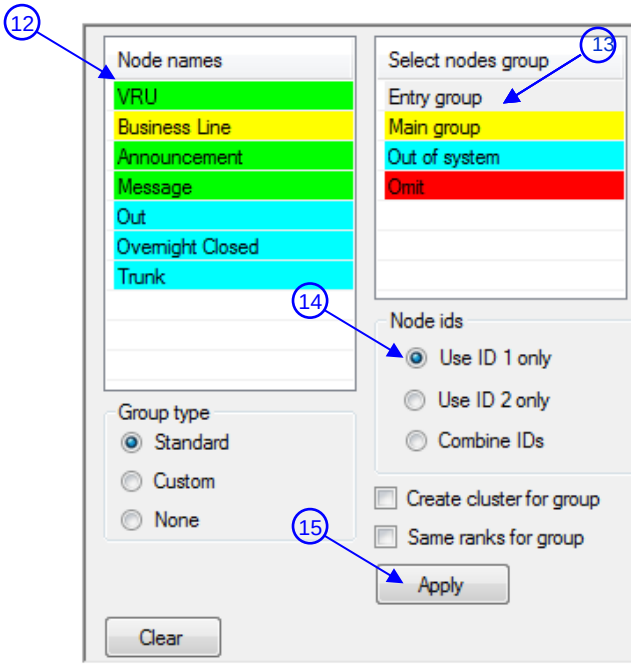
The Flow graph tab:



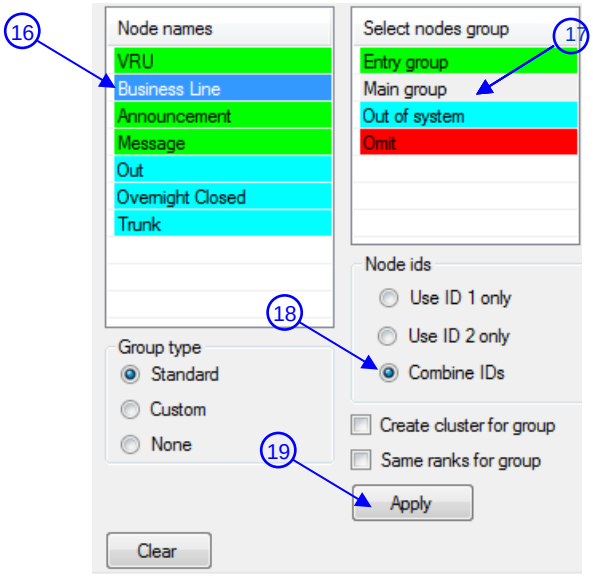
(1) Select path definition: click on the *Select* button (Path 1), select *call_id* field and click OK.
(2) Select *Use layers.*
(3) Define layer (sub path) field: click on the *Select* button, select *cust_subcall* field. Click OK.
(4) Select *Compress layers use first and last record only*
(5) Maximum number of layers = *3*
(6) From Sinks (exits) options select *Use sinks field*.
(7) Define sink field: *outcome* and sink dictionary *flow_outcome*.
(8) Click on the *Mark Sink nodes* button.



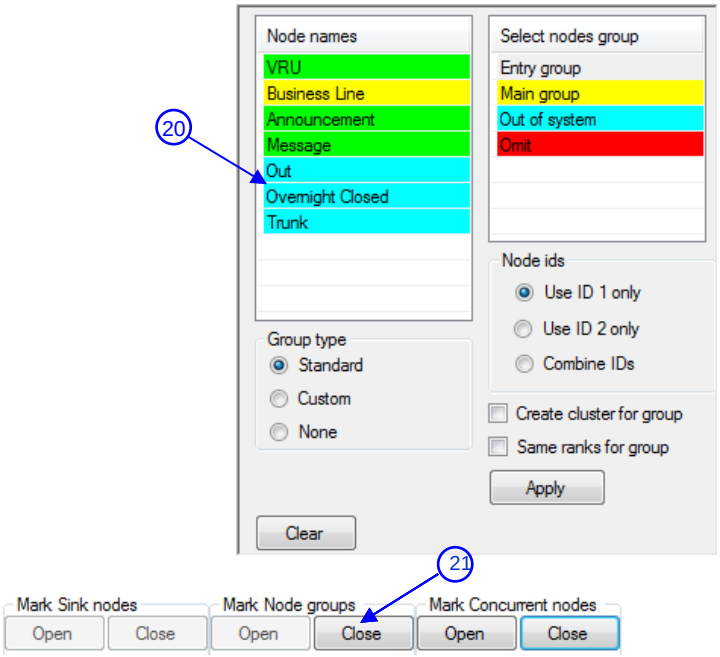
(9) Apply sinks in groups: define *C* as *Normal* group, *A* as *Abnormal* group, and *T* as *Transfer* group.
(10) Close Mark Sink nodes panel.
(11) Open *Mark Node groups* panel



(12) Select nodes: *VRU, Announcement, Message* (on the left side)
(13) Select *Entry group* (on the right side)
(14) Select *Use ID 1 only*.
(15) Click on the *Apply* button.

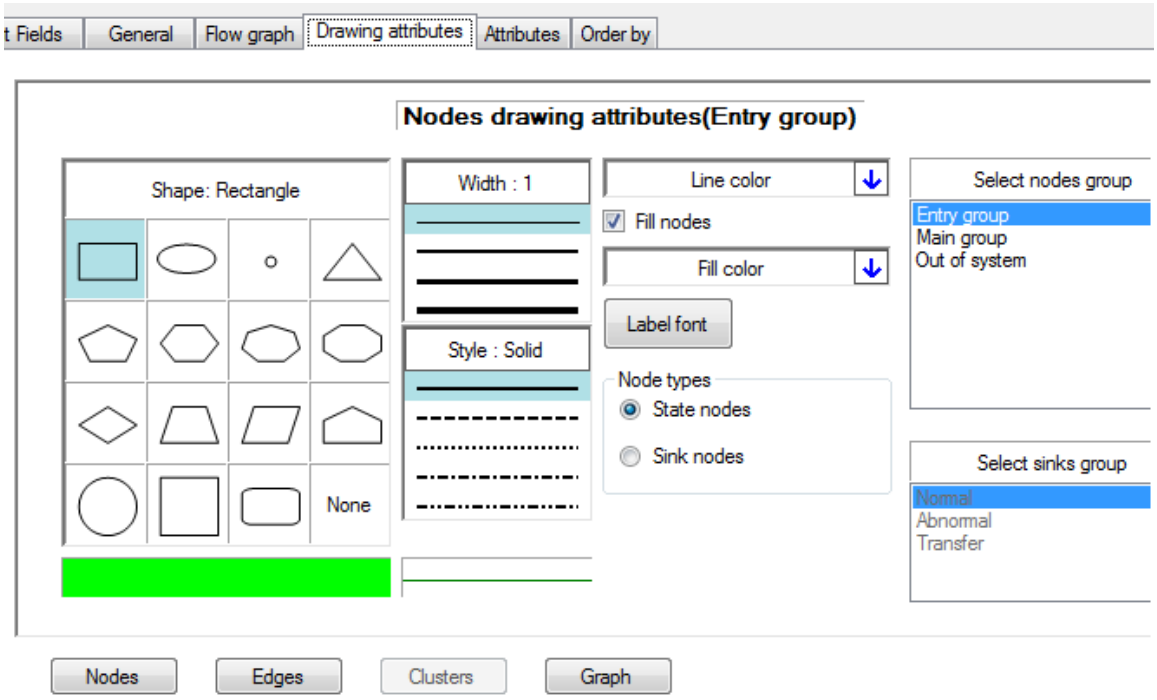


(16) Select *Business Line* node (on the left side)
(17) Select *Main group* (on the **right** side)
(18) Select *Combine IDs*.
(19) Click on the *Apply* button.

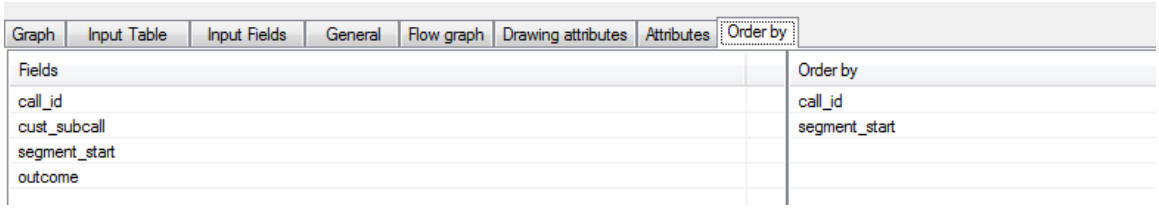


(20) In the same manner, apply the nodes: *Out*, *Overnight Closed*, and *Trunk* to the *Out of system* group, with *Use ID 1 only*.
(21) Close *Mark node groups* panel.

The Drawing attributes tab:

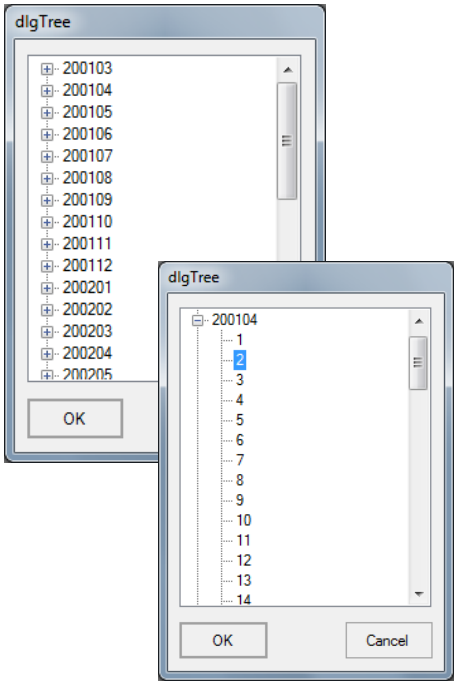


The Order by tab:



Reopen SEEGraph 1.0.
Select *File->Process (one day).* Select the *USBank* study. Click *OK.*
Select *flow_subcalls* graph, click OK.

Select *200104* (April 2001), and select *2* (2 April, 2001), click OK.



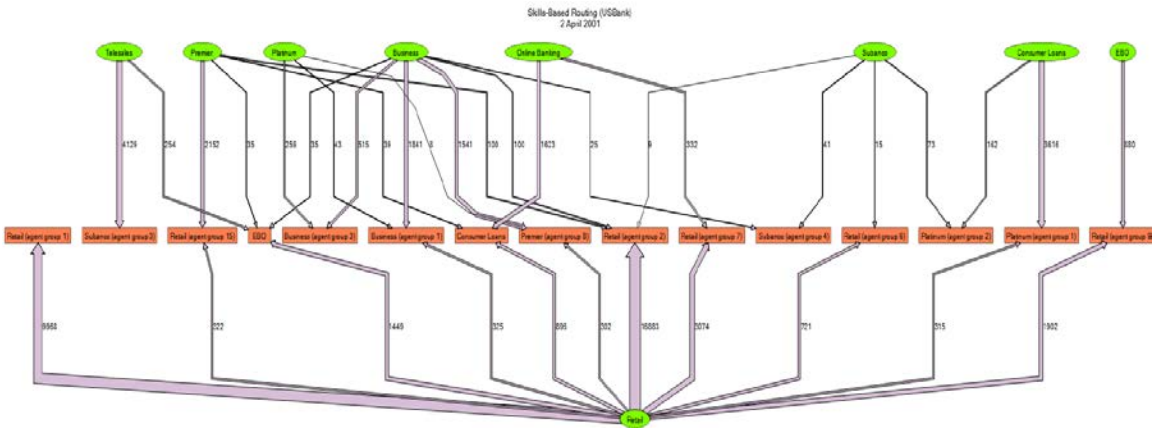
Save file in C:\ animation. Name the text file *flow_subcalls* and click Save.

Open *SEEGraph View 1.0.*Select *File->Open*. Select *flow_subcalls.txt* file from *C:\ animation.* Click on the *Open* button.

Part II Practice

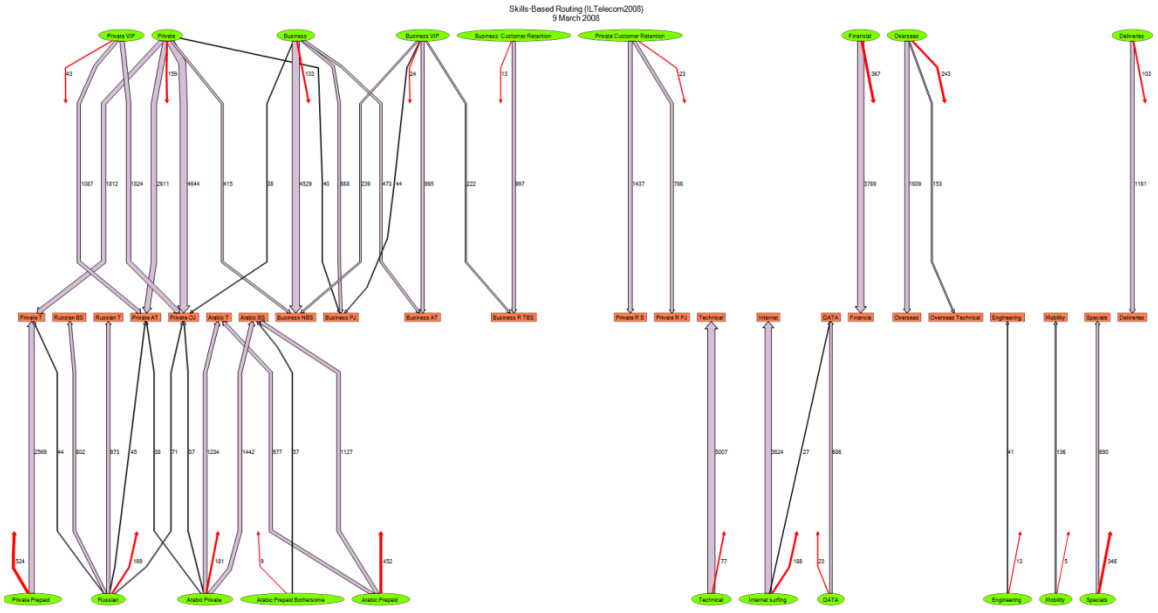
Exercise 1: Create SBR graph

1. Go over the following examples of SBR graphs (ILTelecom2008 data).
2. Read the document on *Skills-Based- Routing in US Bank* *http://ie.technion.ac.il/Labs/Serveng/files/Skills-Based-Routing_USBank.pdf* .
3. Create the following bipartite graph (SBR) by using *USBank* data: without animation, without sinks, balance by main left, from source table cust_subcalls. See Appendix 4 for a description of *cust_subcalls* table.



Examples of design: Skills-Based Routing (SBR) bipartite graph type

1. Graph (without animation)

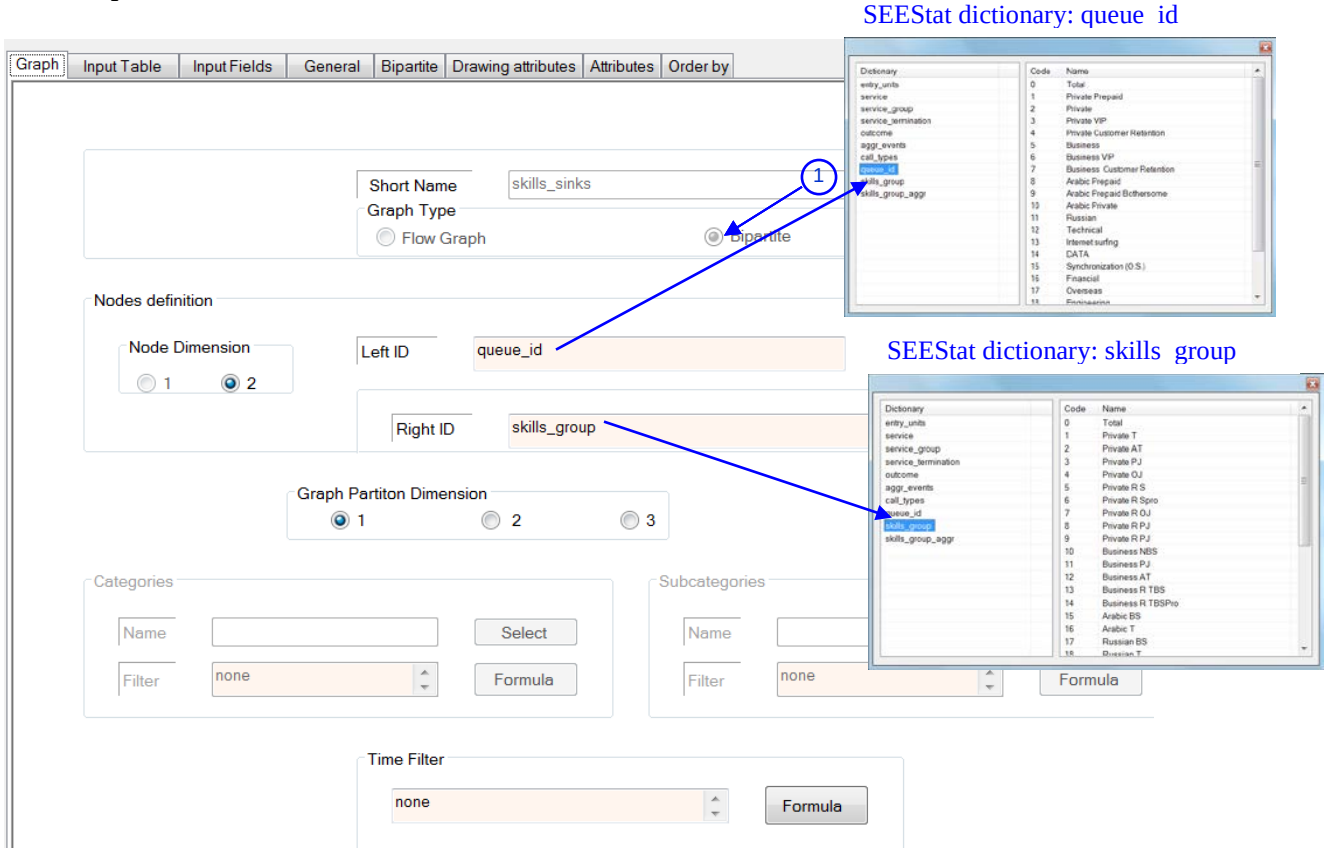


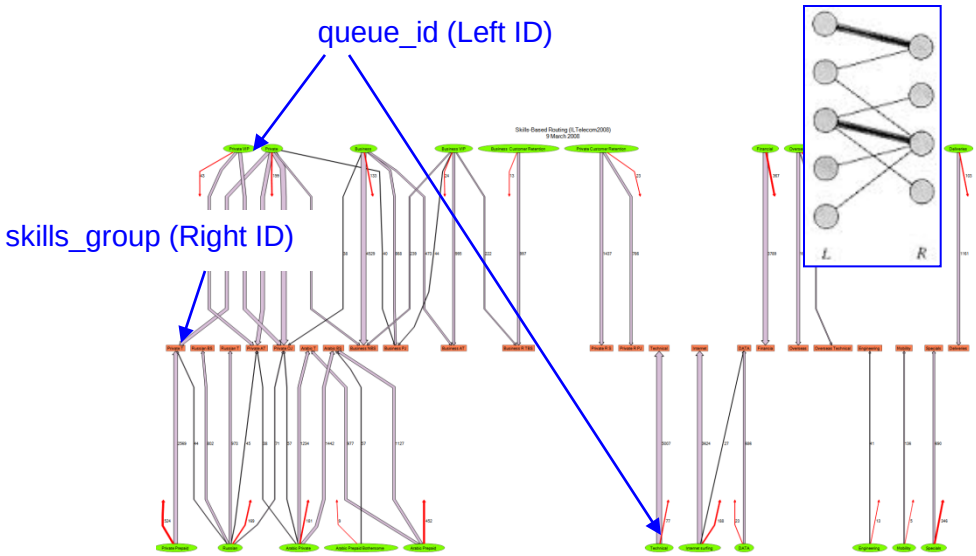
Example of data:

SELECT cust_subcalls.queue_id, cust_subcalls.skills_group, cust_subcalls.service, cust_subcalls.service_time, cust_subcalls.segment_start, cust_subcalls.service_entry, cust_subcalls.segment_end, cust_subcalls.service_group, cust_subcalls.outcome
FROM cust_subcalls
WHERE (((cust_subcalls.queue_id)<99) AND ((cust_subcalls.skills_group)>0 And (cust_subcalls.skills_group)<99) AND ((cust_subcalls.service_group)=2)) OR (((cust_subcalls.queue_id)<99) AND ((cust_subcalls.skills_group)=0) AND ((cust_subcalls.service_group)=2) AND ((cust_subcalls.outcome)>10 And (cust_subcalls.outcome)<20));

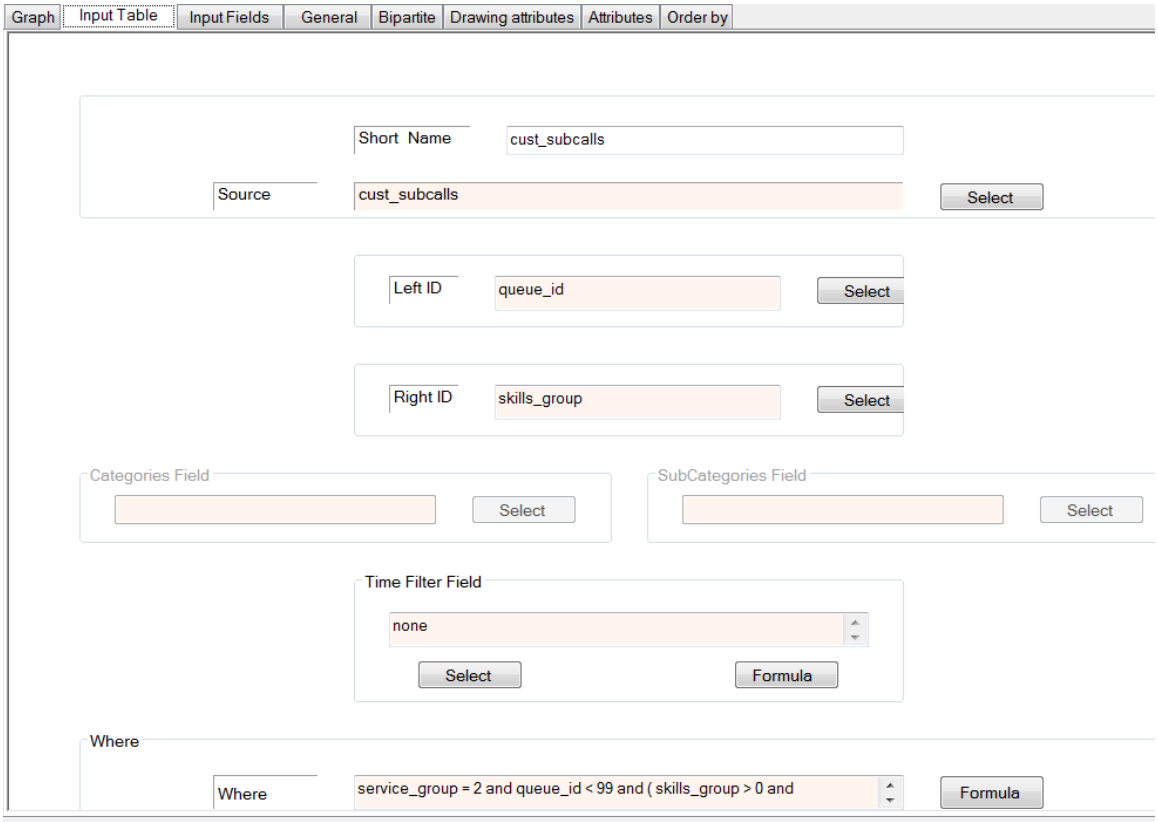
cust_subcalls								
segment_start	service_entry	segment_end	queue_id	skills_group	service	service_time	service_group	outcome
1205020808	1205020843	1205020867	17	24	11	24	2	1
1205020978	1205020989	1205020989	13	0	19	0	2	12
1205021347	1205021353	1205021536	17	24	11	183	2	2
1205021402	1205021563	1205021563	13	0	19	0	2	12
1205021566	1205021566	1205021779	17	24	11	213	2	1
1205021798	1205021828	1205021828	13	0	19	0	2	12
1205023070	1205023071	1205023377	17	24	11	306	2	1
1205023418	1205023424	1205023424	17	0	11	0	2	12
1205024544	1205024632	1205024632	13	0	19	0	2	12

The *Graph* tab:



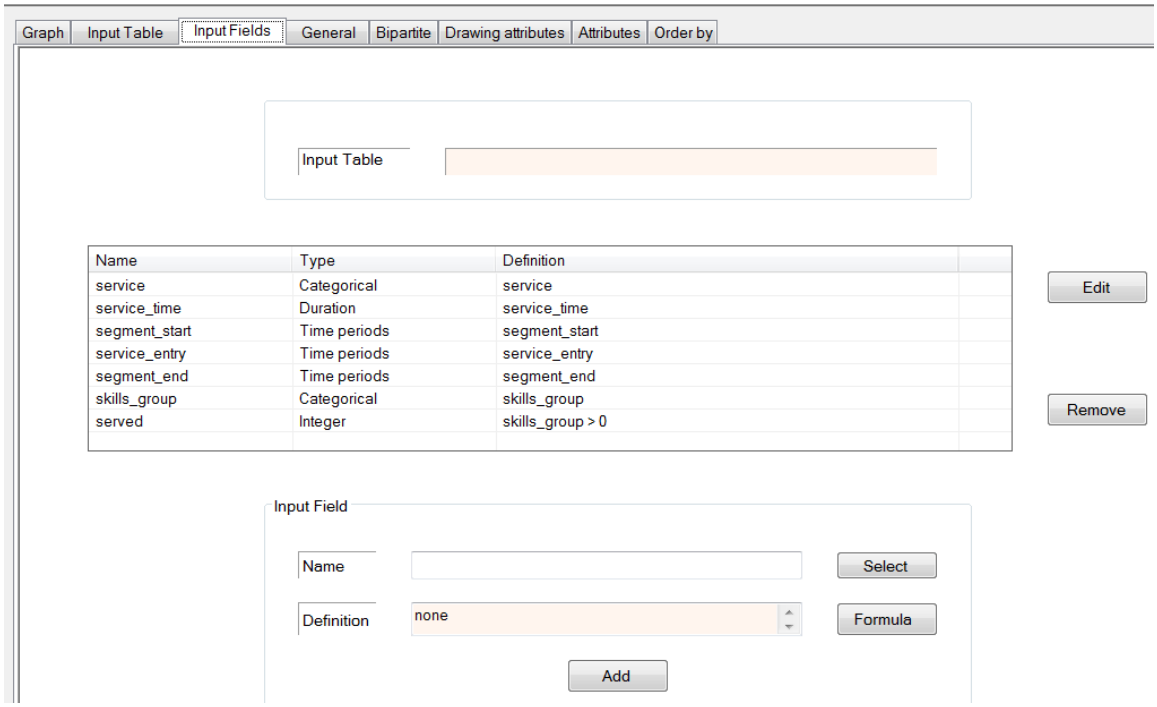


The *Input Table* tab:

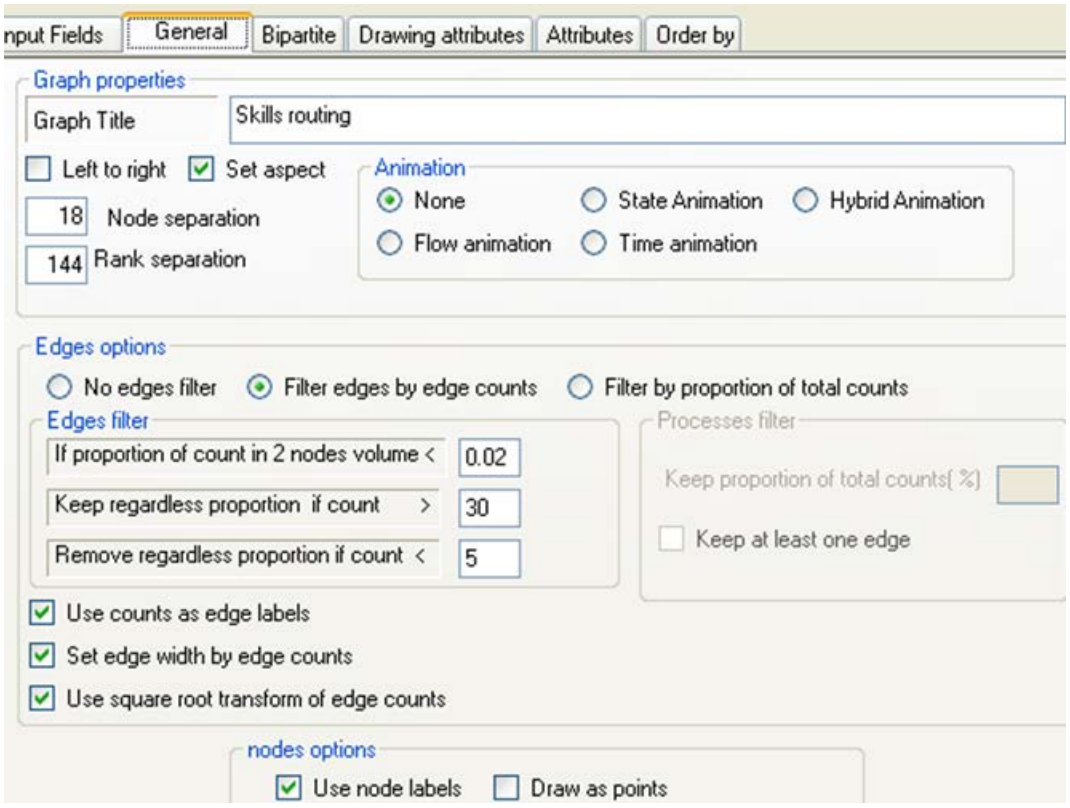



```
service_group = 2 and queue_id < 99 and ( skills_group > 0 and skills_group < 99 or skills_group = 0 and outcome >
10 and outcome < 20 )
```

The *Input Fields* tab:

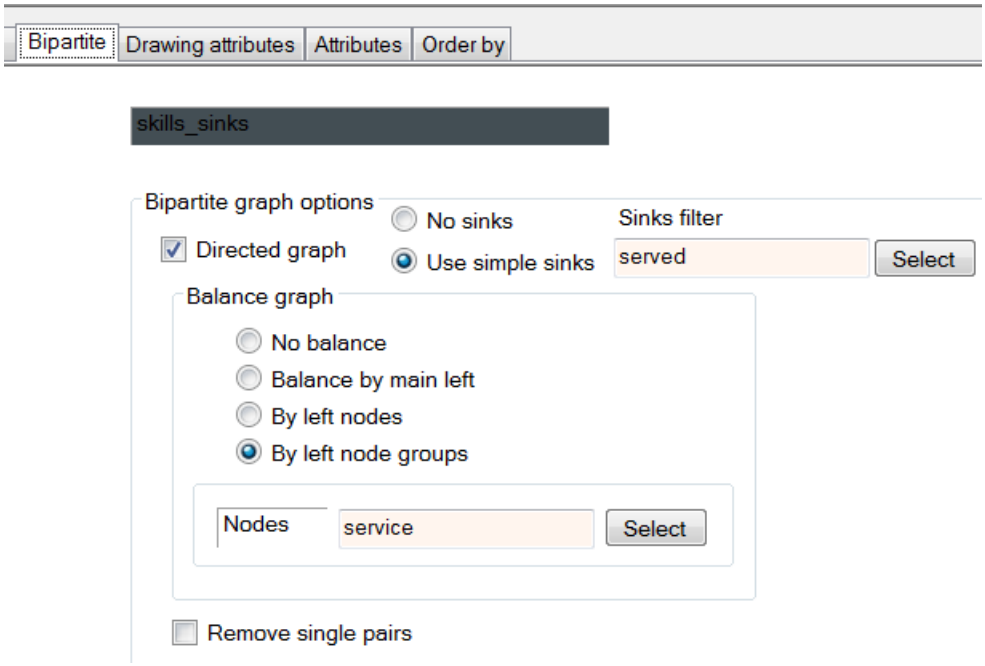


The *General* tab:

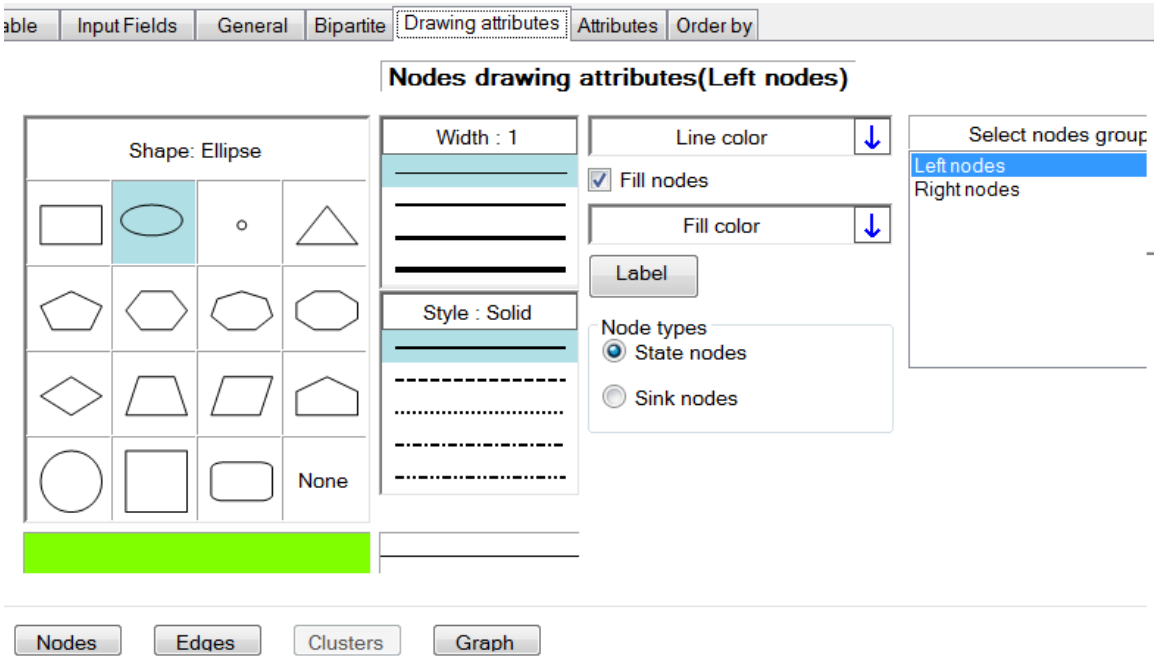


The *Bipartite* tab:

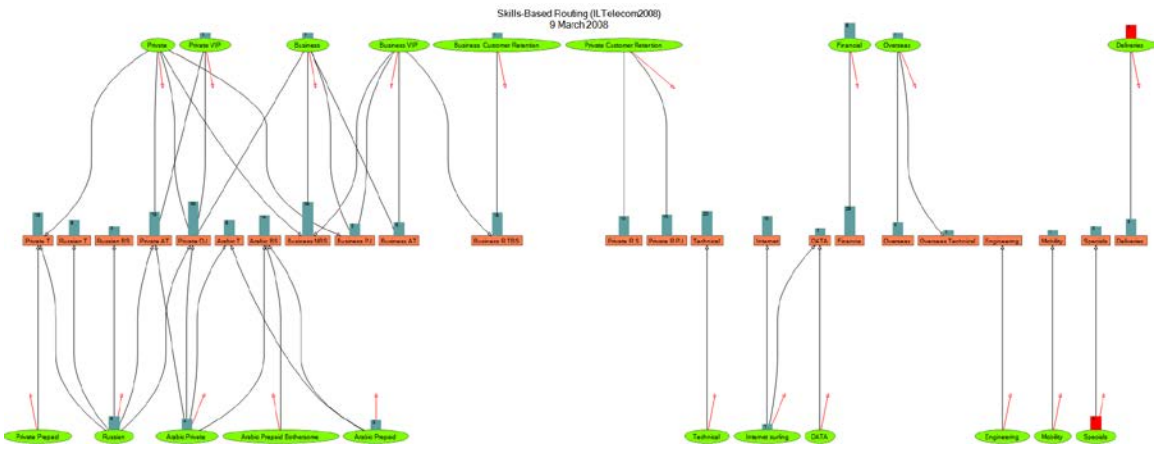
See detailed description of *Bipartite* tab in Appendix 2.

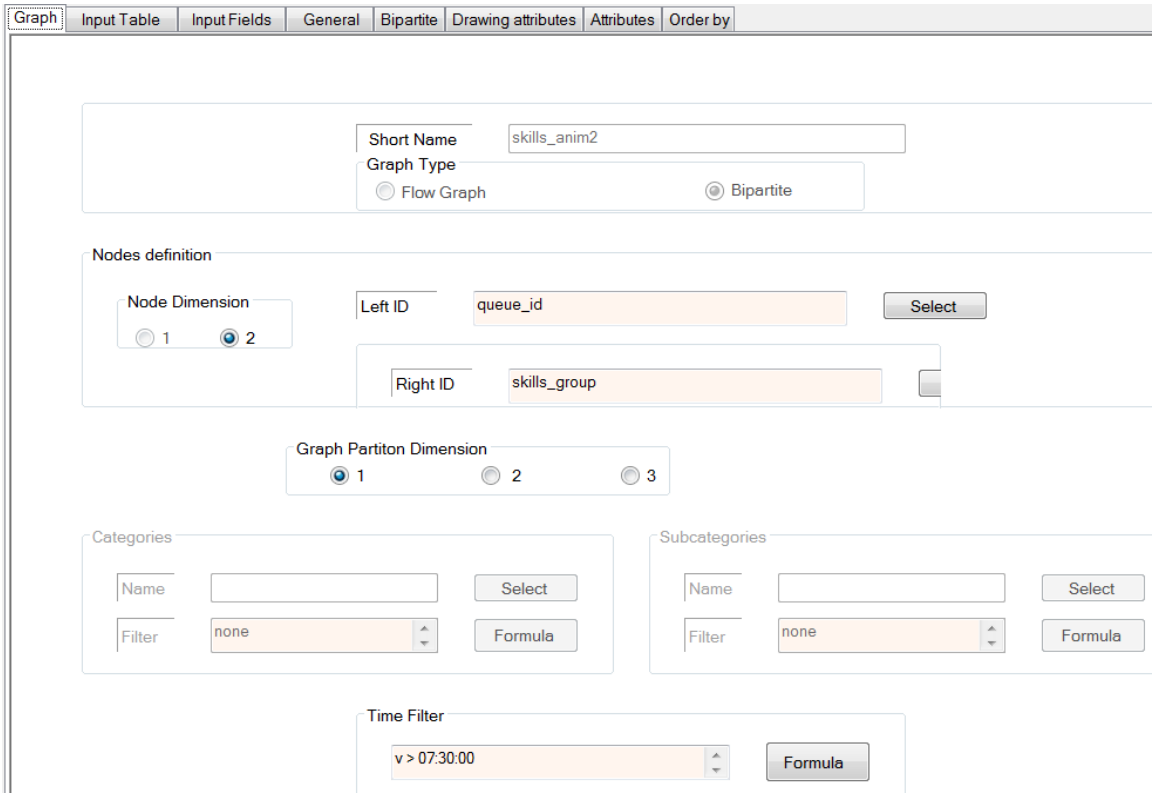


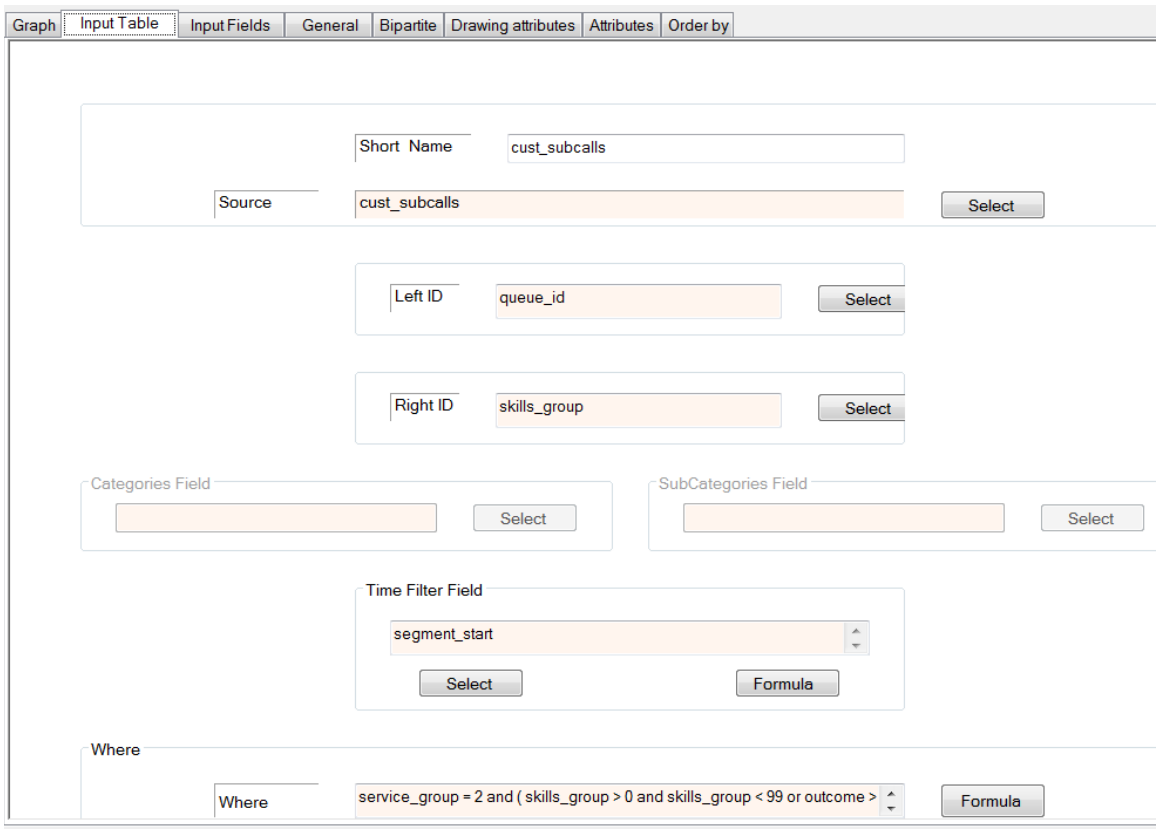
The *Drawing attributes* tab:

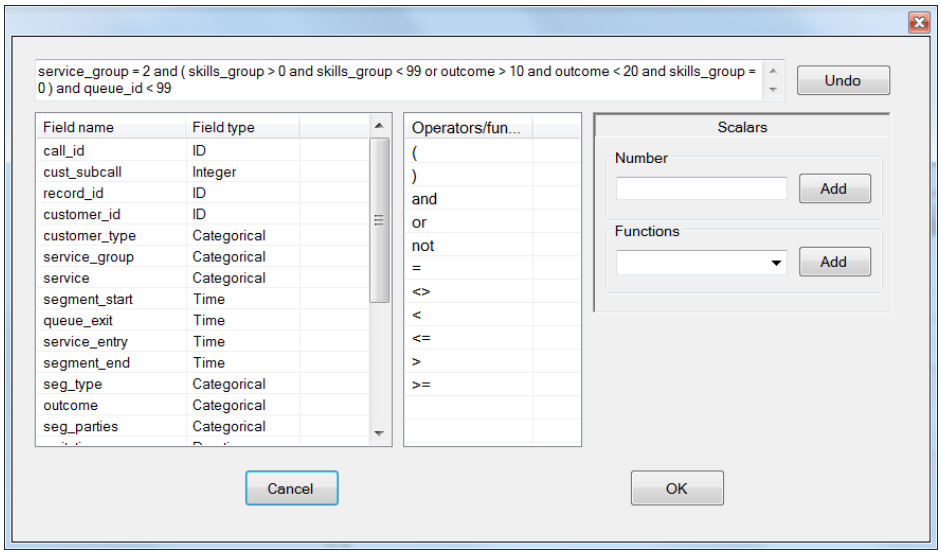


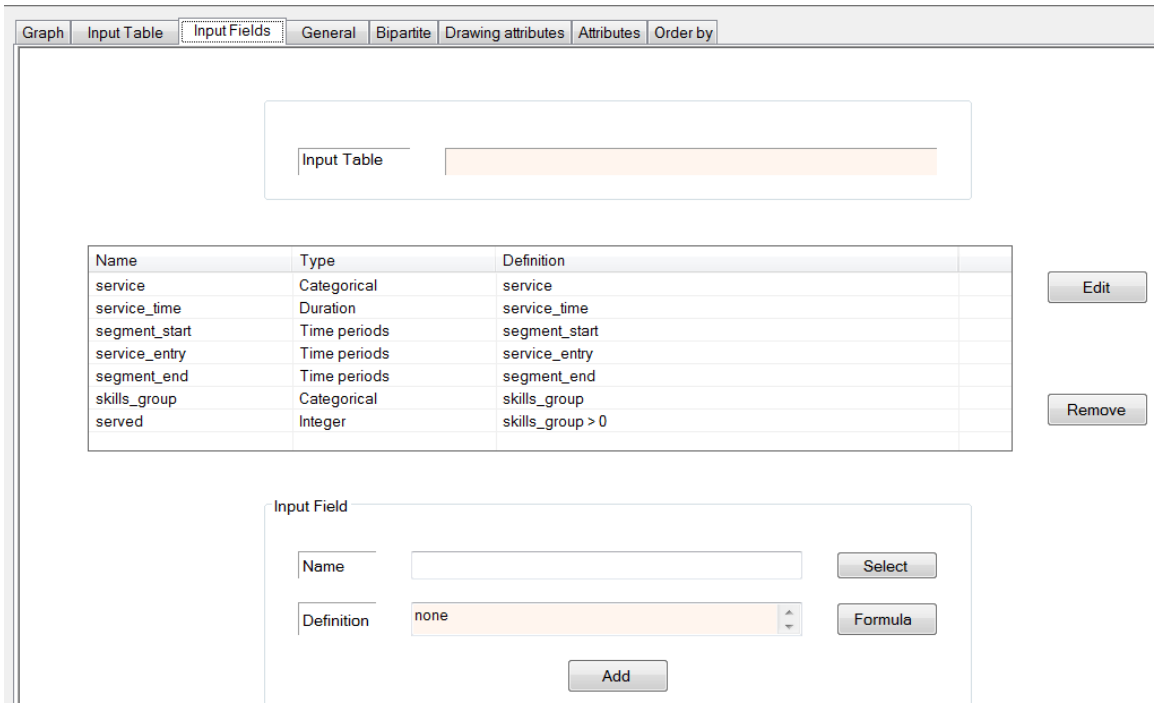
2. State animation: Operations Research (or queueing) view



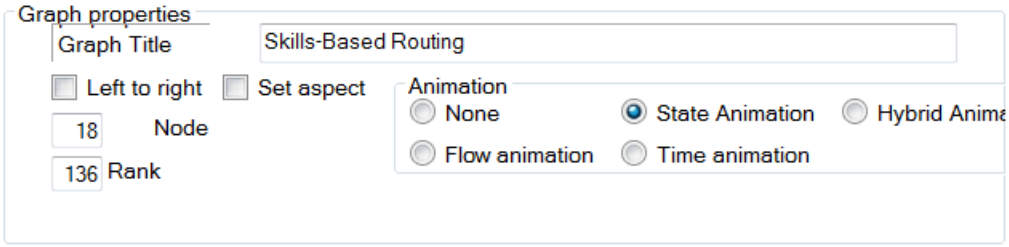


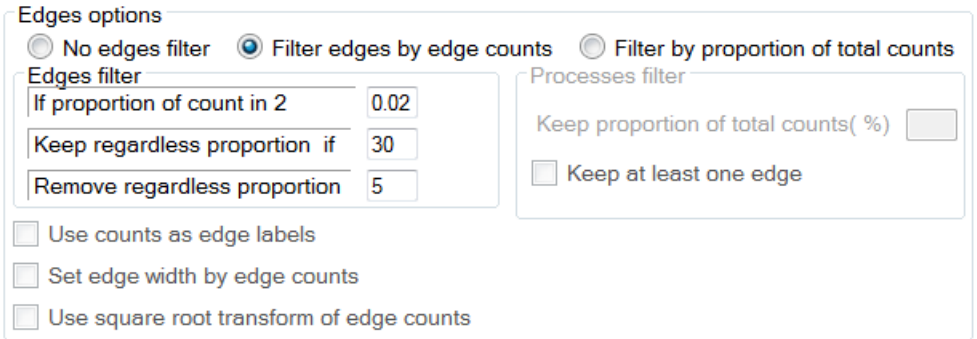

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Short Name
cust_subcalls
Source
cust_subcalls
Select
Left ID
queue_id
Select
Right ID
skills_group
Select
Categories Field
Select
SubCategories Field
Select
Time Filter Field
segment_start
Select
Formula
Where
Where
service_group = 2 and (skills_group > 0 and skills_group < 99 or outcome >
Formula


service_group = 2 and (skills_group > 0 and skills_group < 99 or outcome > 10 and outcome < 20 and skills_group = 0) and queue_id < 99
Undo
Field name
Field type
call_id ID
cust_subcall Integer
record_id ID
customer_id ID
customer_type Categorical
service_group Categorical
service Categorical
segment_start Time
queue_exit Time
service_entry Time
segment_end Time
seg_type Categorical
outcome Categorical
seg_parties Categorical
Operators/fun...
(
)
and
or
not
=
<>
<
<=
>
>=
Scalars
Number
Add
Functions
Add
Cancel
OK

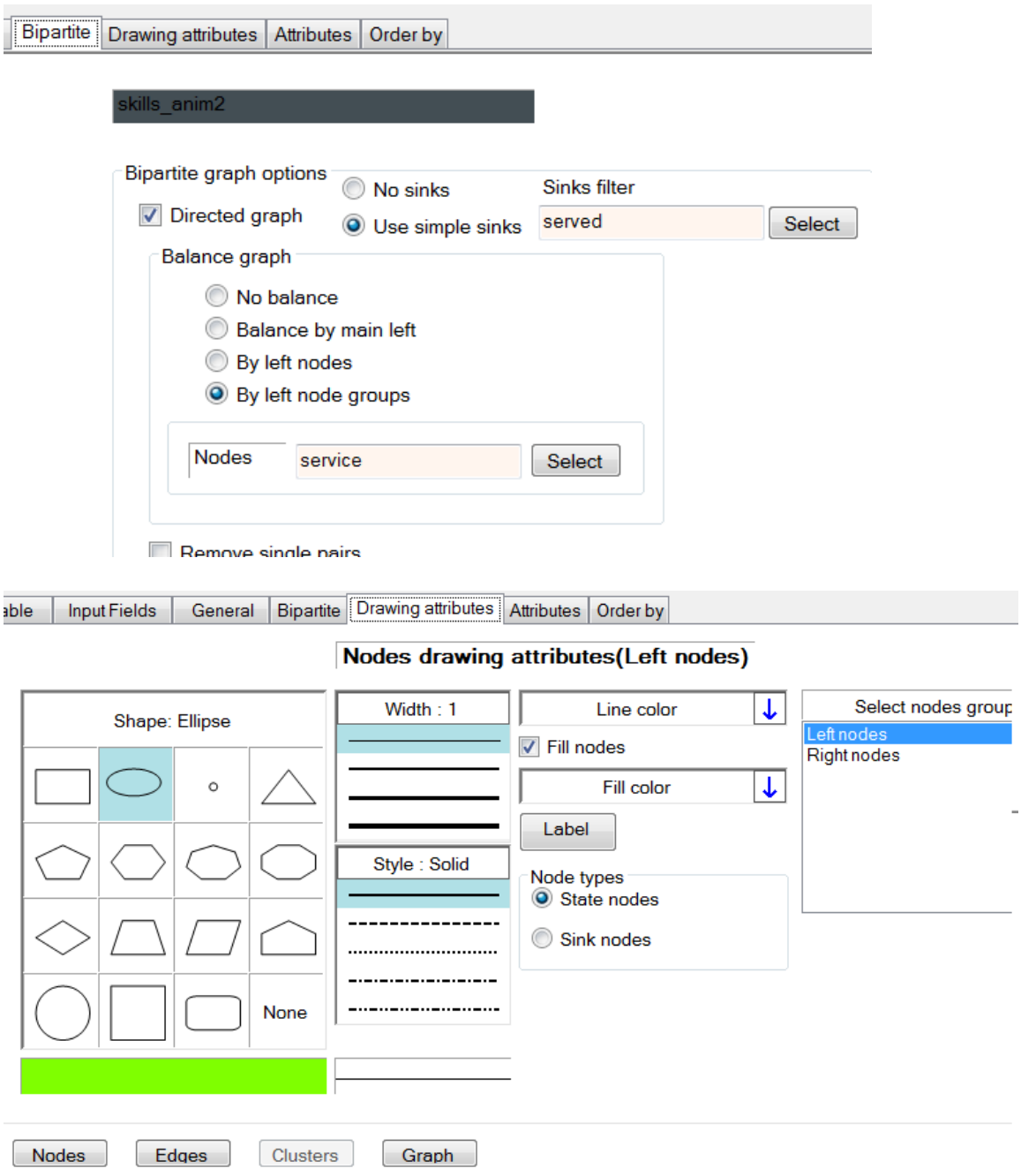

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Input Table
Name
Type
Definition
service
Categorical
service
service_time
Duration
service_time
segment_start
Time periods
segment_start
service_entry
Time periods
service_entry
segment_end
Time periods
segment_end
skills_group
Categorical
skills_group
served
Integer
skills_group > 0
Edit
Remove
Input Field
Name
Select
Definition
none
Formula
Add


ds
General
Bipartite
Drawing attributes
Attributes
Order by

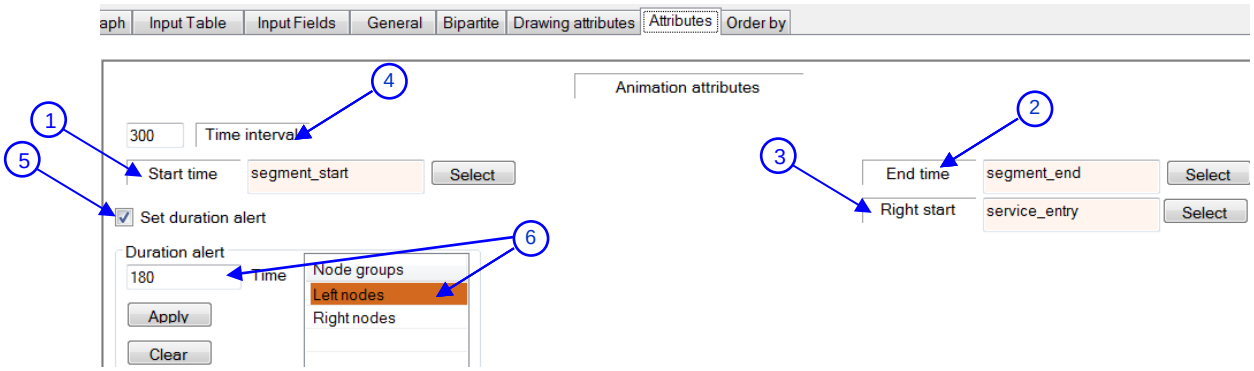

Graph properties
Graph Title
Skills-Based Routing
Left to right
Set aspect
18
Node
136
Rank
Animation
None
State Animation
Hybrid Anima
Flow animation
Time animation


Edges options
No edges filter
Filter edges by edge counts
Filter by proportion of total counts
Edges filter
If proportion of count in 2
0.02
Keep regardless proportion if
30
Remove regardless proportion
5
Processes filter
Keep proportion of total counts(%)
Keep at least one edge
Use counts as edge labels
Set edge width by edge counts
Use square root transform of edge counts


nodes options
Use node labels
Draw as points

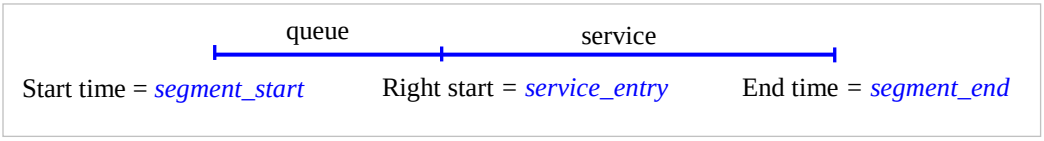

Bipartite
Drawing attributes
Attributes
Order by
skills_anim2
Bipartite graph options
Directed graph
No sinks
Use simple sinks
Sinks filter
served
Select
Balance graph
No balance
Balance by main left
By left nodes
By left node groups
Nodes
service
Select
Remove single pairs
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Nodes drawing attributes(Left nodes)
Shape: Ellipse
None
Width : 1
Style : Solid
Line color
Fill nodes
Fill color
Label
Node types
State nodes
Sink nodes
Select nodes group
Left nodes
Right nodes
Nodes
Edges
Clusters
Graph

The *Attributes* tab:

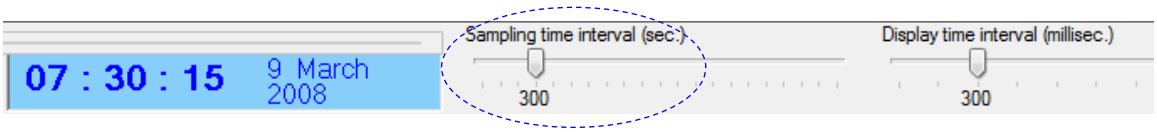


Animation attributes:

(1) Start time = *segment_start*
(2) End time = *segment_end*
(3) Right start = *service_entry*

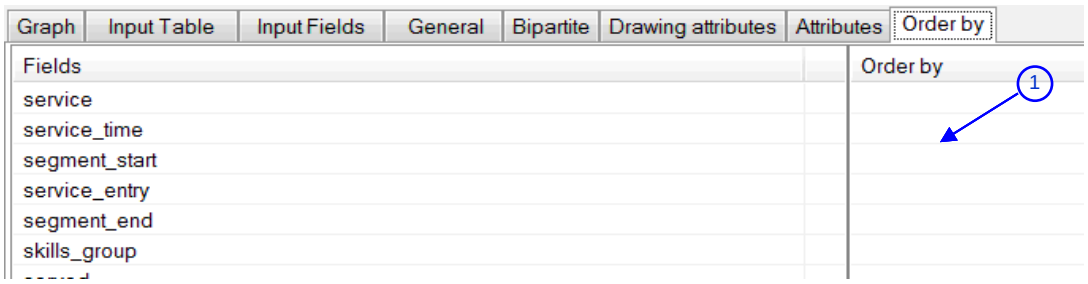


(4) Time interval = *300* – default value for time animation



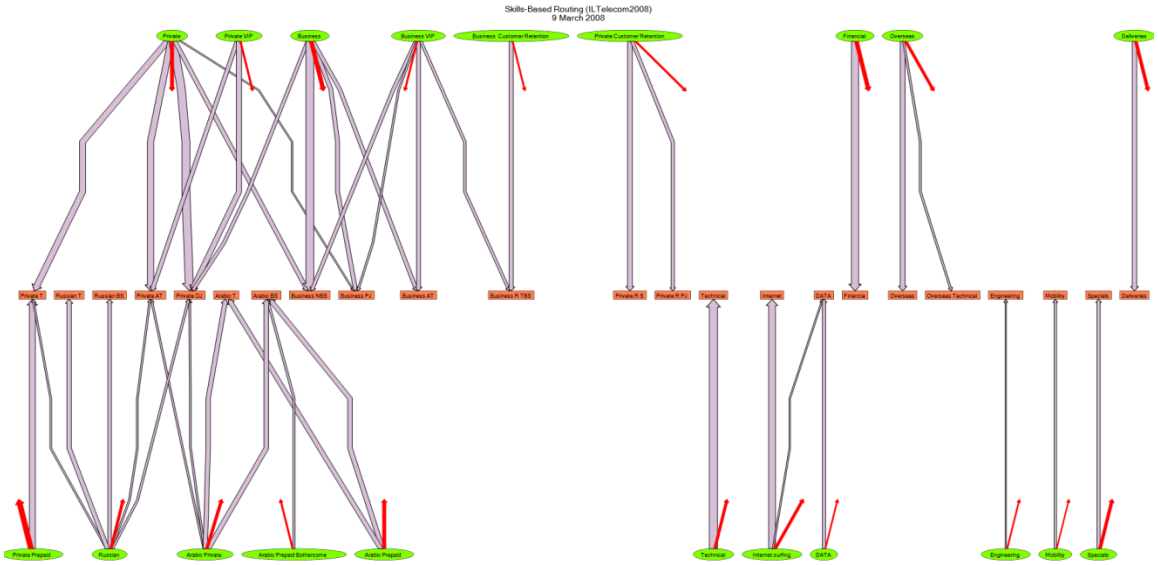
(5) Set duration alert:
Duration alert = *180* – if the **maximum** waiting time, among all the calls in queue at the current moment, exceeds 3 minutes, give an alert by red color.

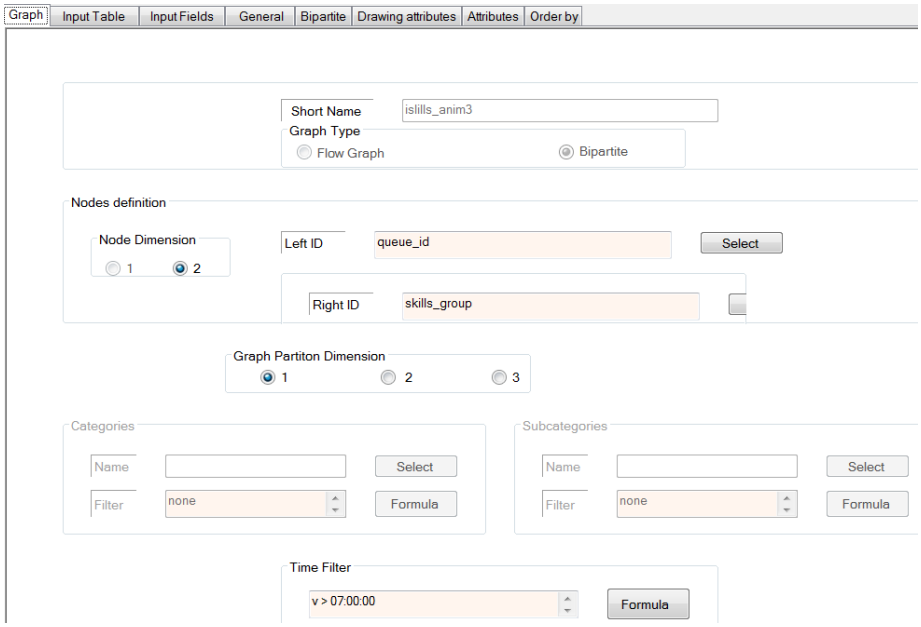
(6) Node groups = *Left nodes* – gives alert according to the measured waiting time in **queue** (in this example 180 seconds).
If Node groups = *Right nodes* – gives alert according to the measured **service** time (not relevant for this case).

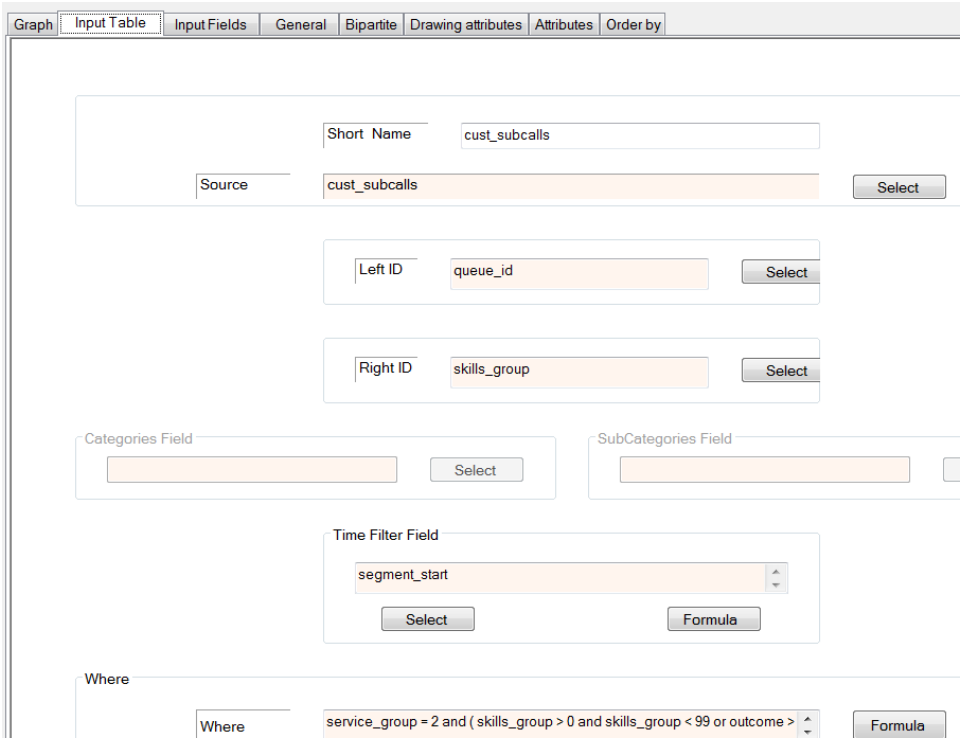


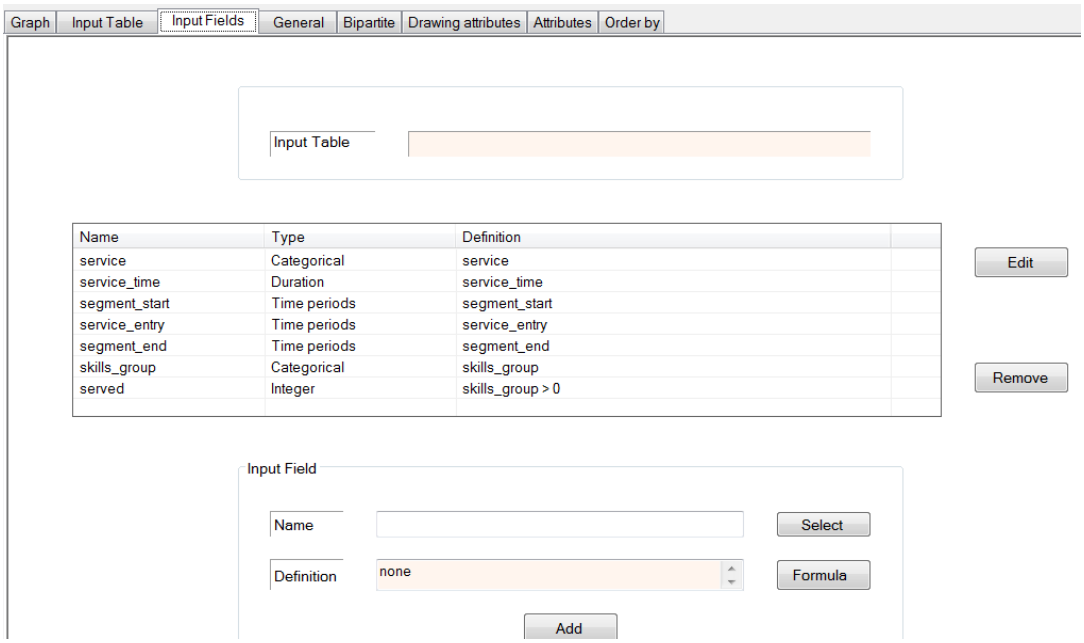
(1) In bipartite graphs there is no need to select *Order by* fields.

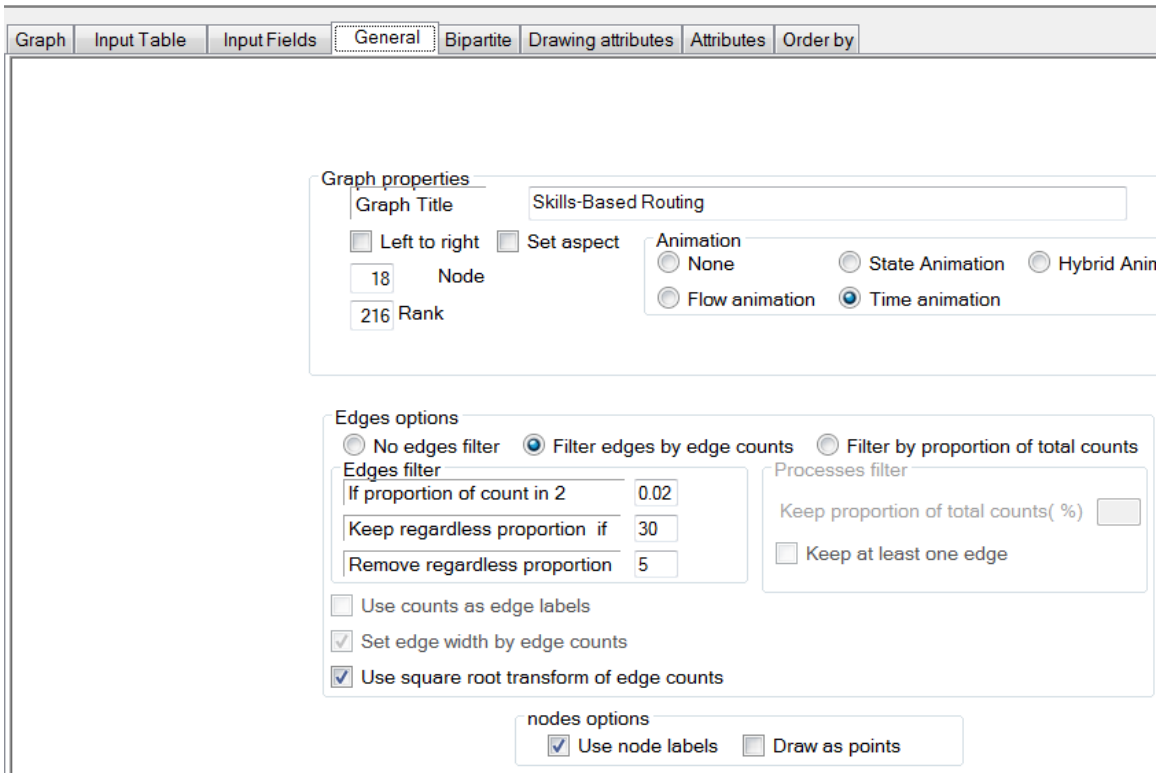
3. Time animation: Network (structure) view

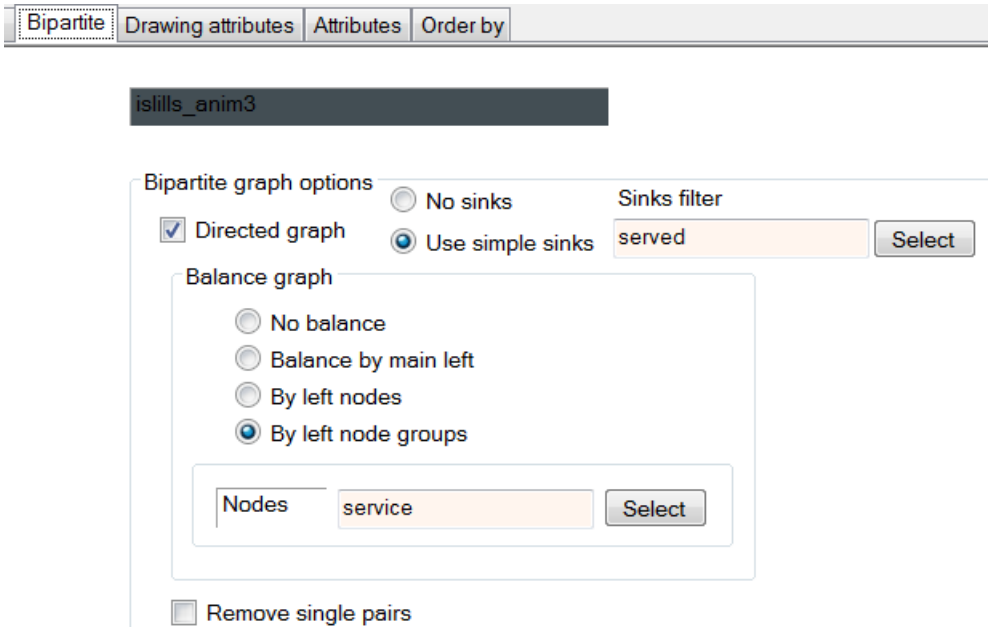


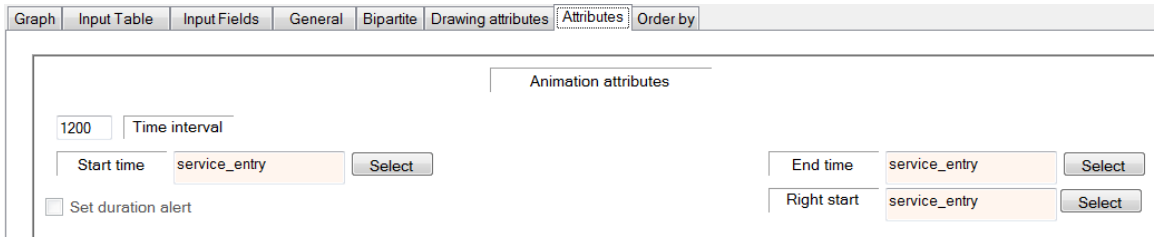



Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Short Name
cust_subcalls
Source
cust_subcalls
Select
Left ID
queue_id
Select
Right ID
skills_group
Select
Categories Field
Select
SubCategories Field
Time Filter Field
segment_start
Select
Formula
Where
Where
service_group = 2 and (skills_group > 0 and skills_group < 99 or outcome >
Formula

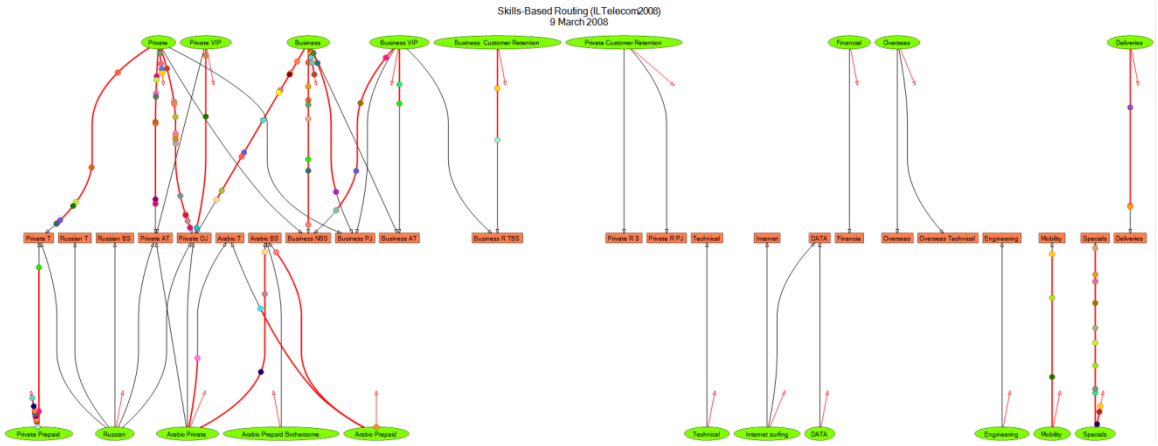

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Input Table
Name
Type
Definition
service
Categorical
service
service_time
Duration
service_time
segment_start
Time periods
segment_start
service_entry
Time periods
service_entry
segment_end
Time periods
segment_end
skills_group
Categorical
skills_group
served
Integer
skills_group > 0
Edit
Remove
Input Field
Name
Select
Definition
none
Formula
Add

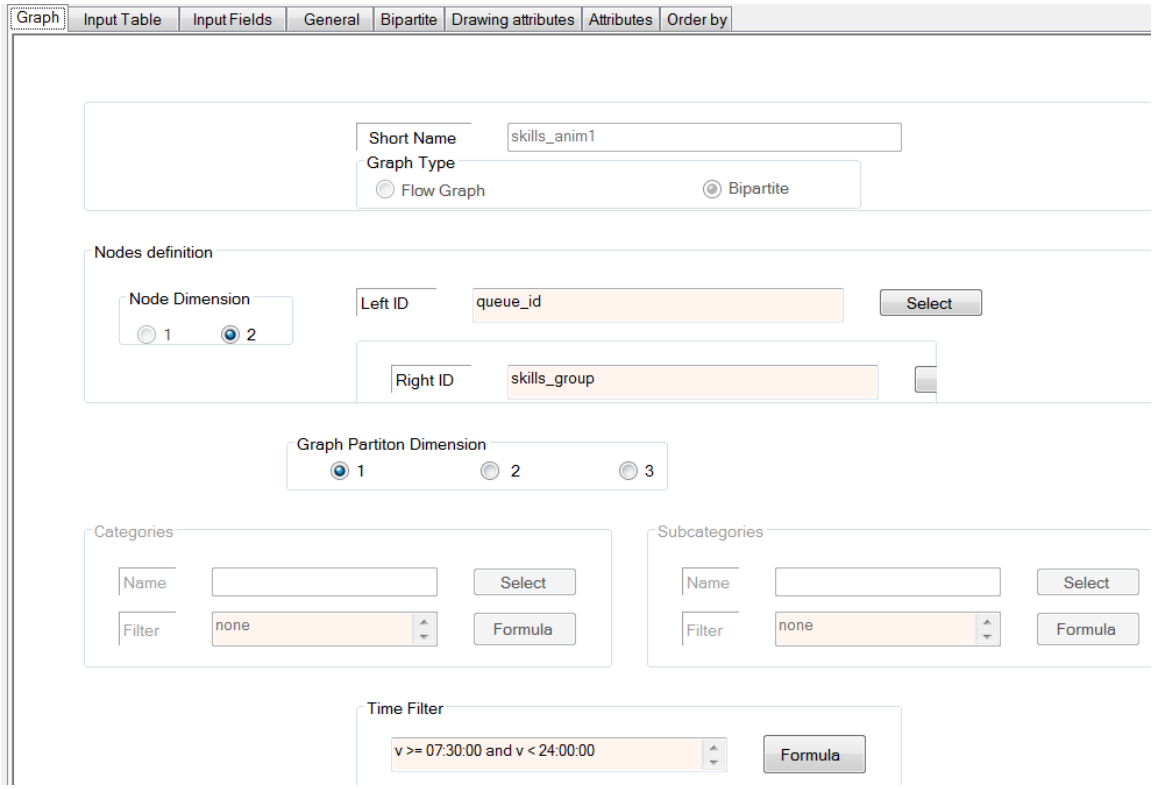

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Graph properties
Graph Title
Skills-Based Routing
Left to right
Set aspect
18
Node
216
Rank
Animation
None
State Animation
Hybrid Anim
Flow animation
Time animation
Edges options
No edges filter
Filter edges by edge counts
Filter by proportion of total counts
Edges filter
If proportion of count in 2
0.02
Keep regardless proportion if
30
Remove regardless proportion
5
Processes filter
Keep proportion of total counts(%)
Keep at least one edge
Use counts as edge labels
Set edge width by edge counts
Use square root transform of edge counts
nodes options
Use node labels
Draw as points

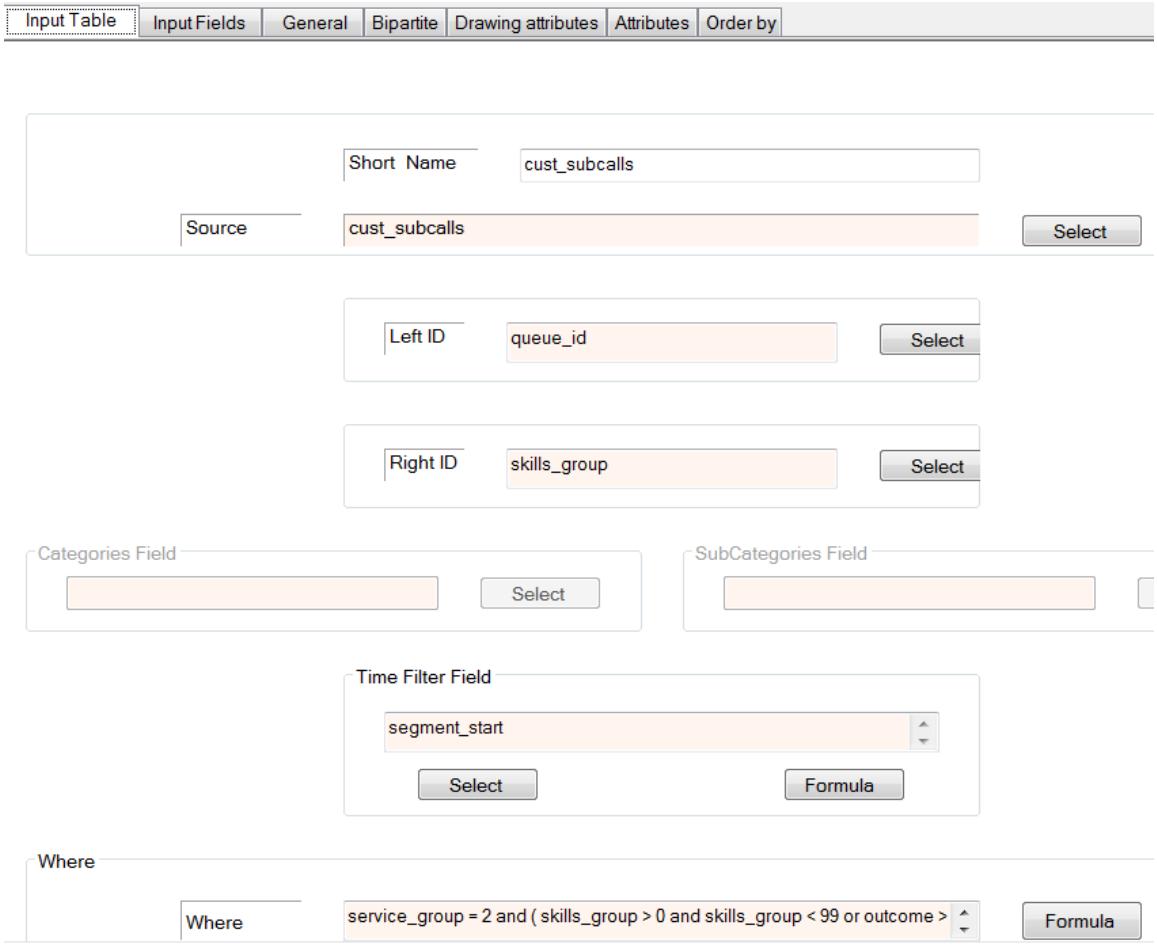

Bipartite
Drawing attributes
Attributes
Order by
islills_anim3
Bipartite graph options
Directed graph
No sinks
Use simple sinks
Sinks filter
served
Select
Balance graph
No balance
Balance by main left
By left nodes
By left node groups
Nodes
service
Select
Remove single pairs

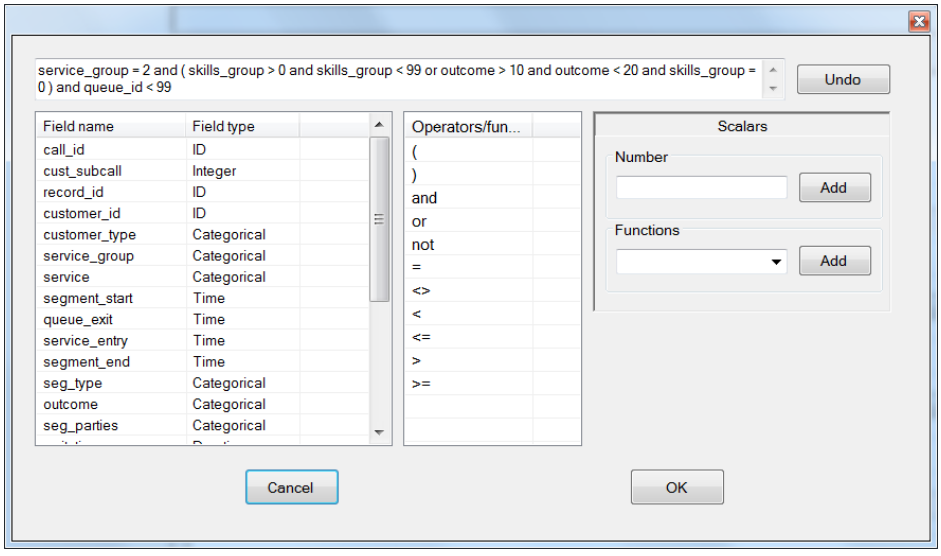

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Animation attributes
1200
Time interval
Start time
service_entry
Select
End time
service_entry
Select
Right start
service_entry
Select
Set duration alert

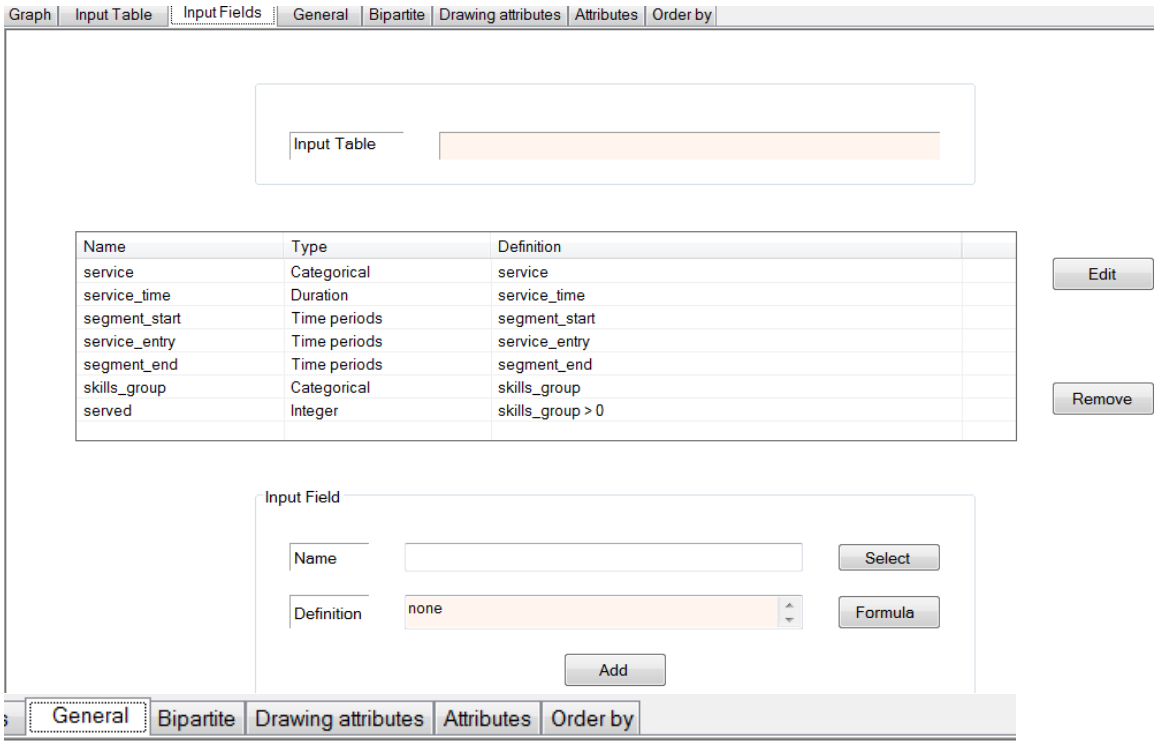
4. Flow animation: Process Mining view

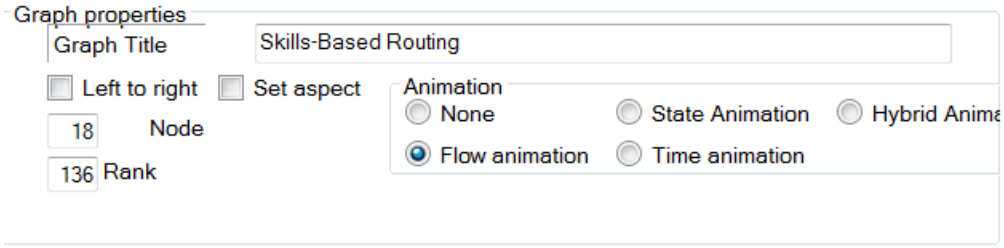


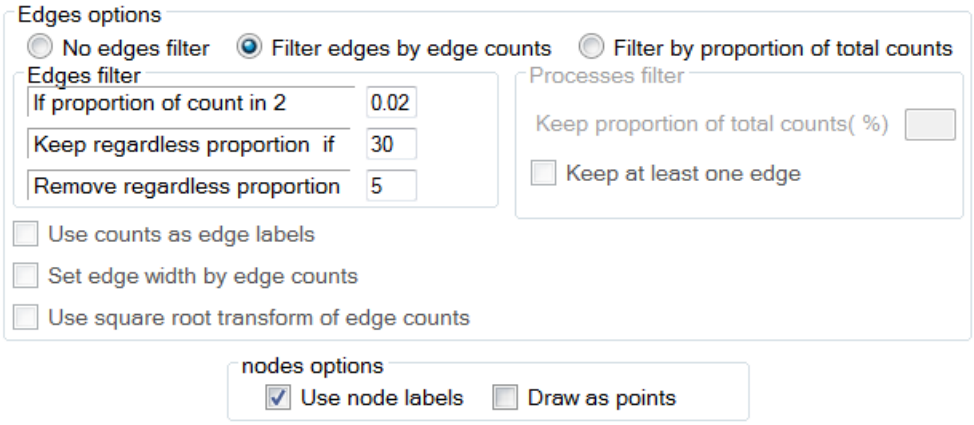


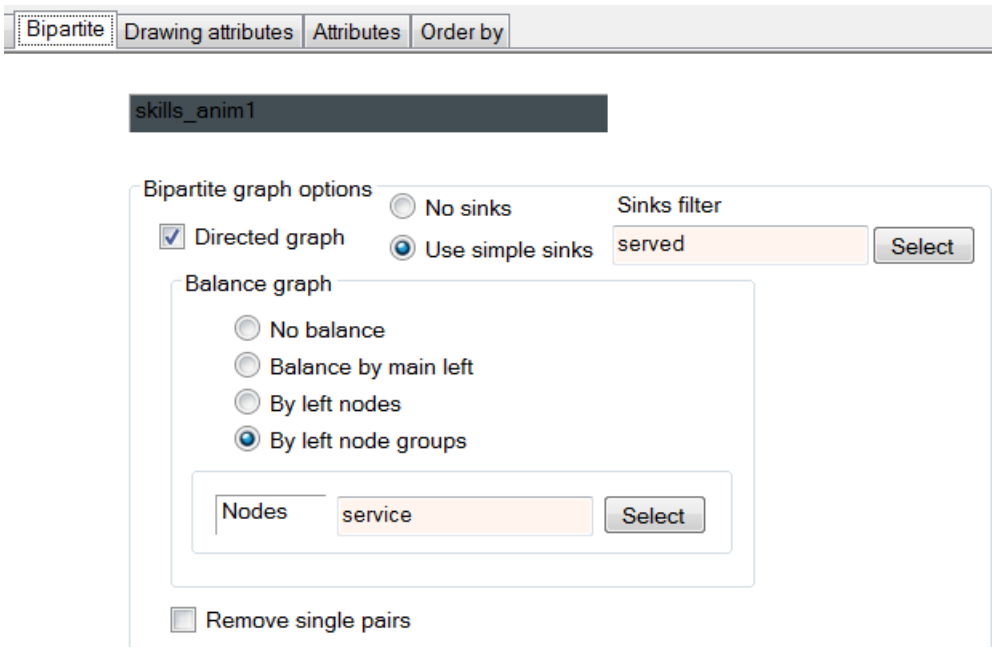

Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Short Name
cust_subcalls
Source
cust_subcalls
Select
Left ID
queue_id
Select
Right ID
skills_group
Select
Categories Field
Select
SubCategories Field
Time Filter Field
segment_start
Select
Formula
Where
Where
service_group = 2 and (skills_group > 0 and skills_group < 99 or outcome >
Formula

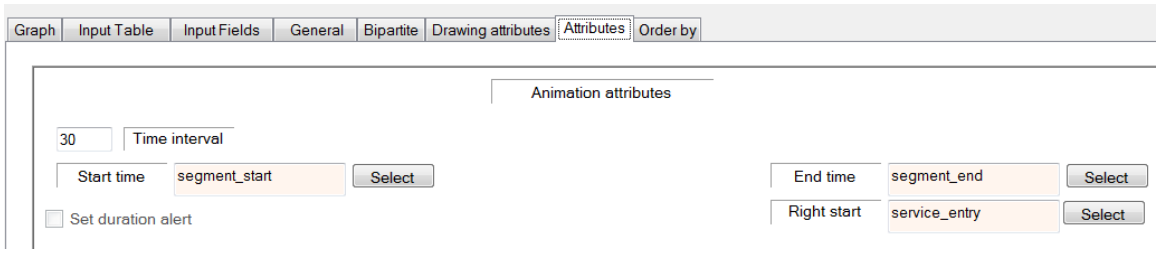

service_group = 2 and (skills_group > 0 and skills_group < 99 or outcome > 10 and outcome < 20 and skills_group = 0) and queue_id < 99
Undo
Field name
Field type
call_id ID
cust_subcall Integer
record_id ID
customer_id ID
customer_type Categorical
service_group Categorical
service Categorical
segment_start Time
queue_exit Time
service_entry Time
segment_end Time
seg_type Categorical
outcome Categorical
seg_parties Categorical
Operators/fun...
(
)
and
or
not
=
<>
<
<=
>
>=
Scalars
Number
Add
Functions
Add
Cancel
OK


Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Input Table
Name
Type
Definition
service
Categorical
service
service_time
Duration
service_time
segment_start
Time periods
segment_start
service_entry
Time periods
service_entry
segment_end
Time periods
segment_end
skills_group
Categorical
skills_group
served
Integer
skills_group > 0
Edit
Remove
Input Field
Name
Select
Definition
none
Formula
Add
General
Bipartite
Drawing attributes
Attributes
Order by

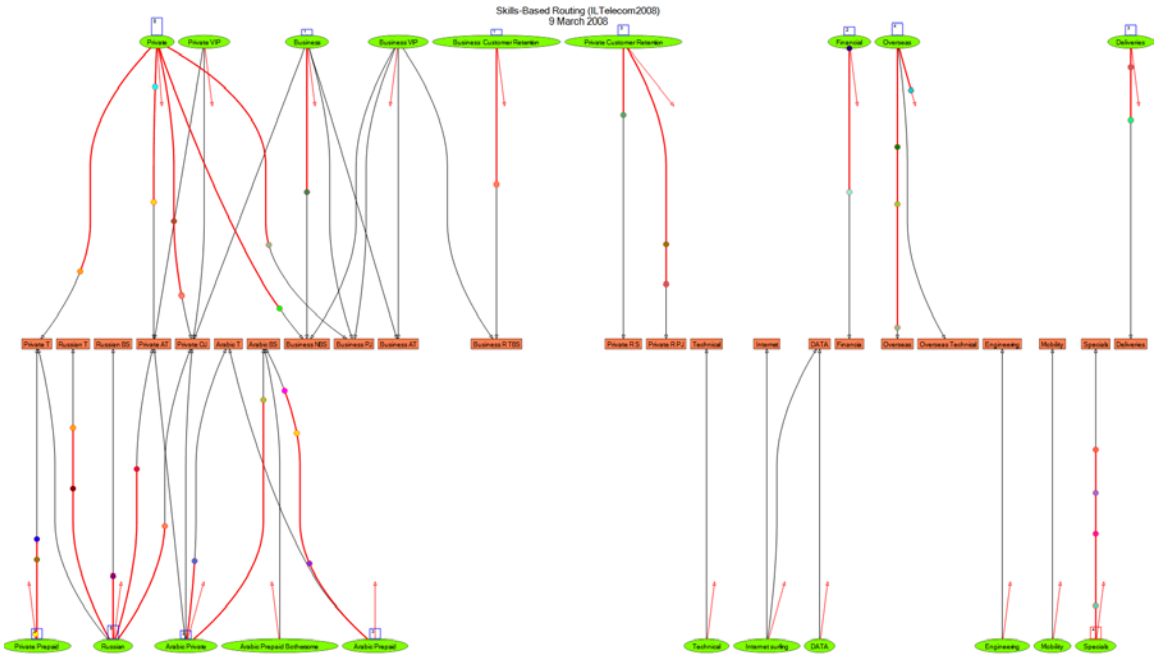

Graph properties
Graph Title
Skills-Based Routing
Left to right
Set aspect
18
Node
136
Rank
Animation
None
State Animation
Hybrid Anima
Flow animation
Time animation

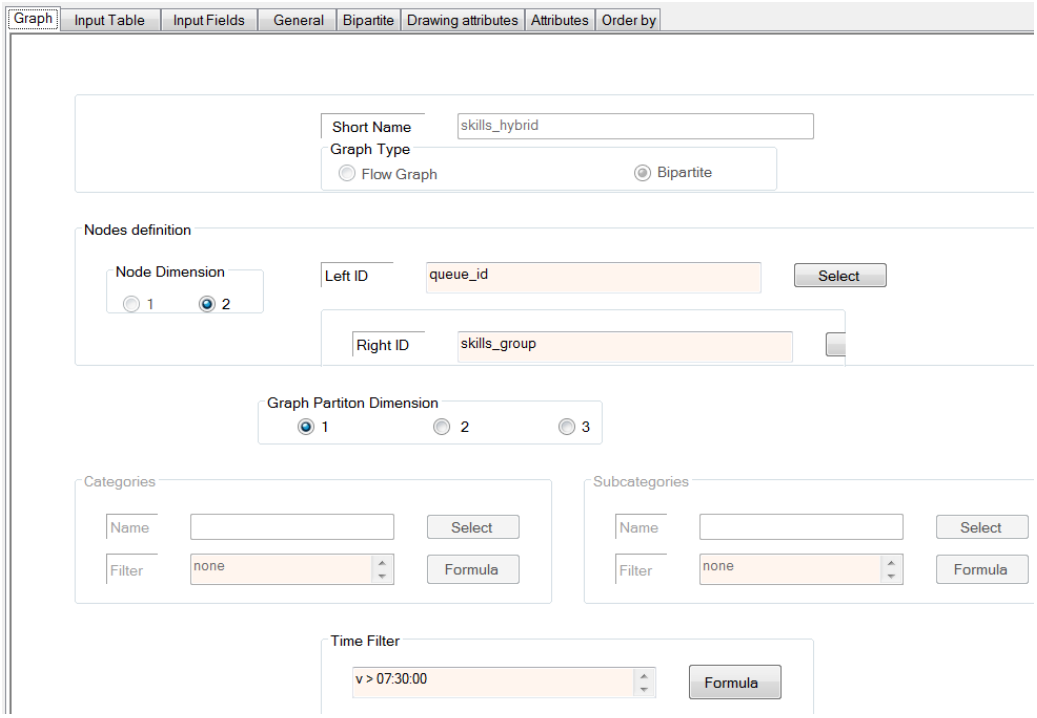

Edges options
No edges filter
Filter edges by edge counts
Filter by proportion of total counts
Edges filter
If proportion of count in 2
0.02
Keep regardless proportion if
30
Remove regardless proportion
5
Processes filter
Keep proportion of total counts(%)
Keep at least one edge
Use counts as edge labels
Set edge width by edge counts
Use square root transform of edge counts
nodes options
Use node labels
Draw as points

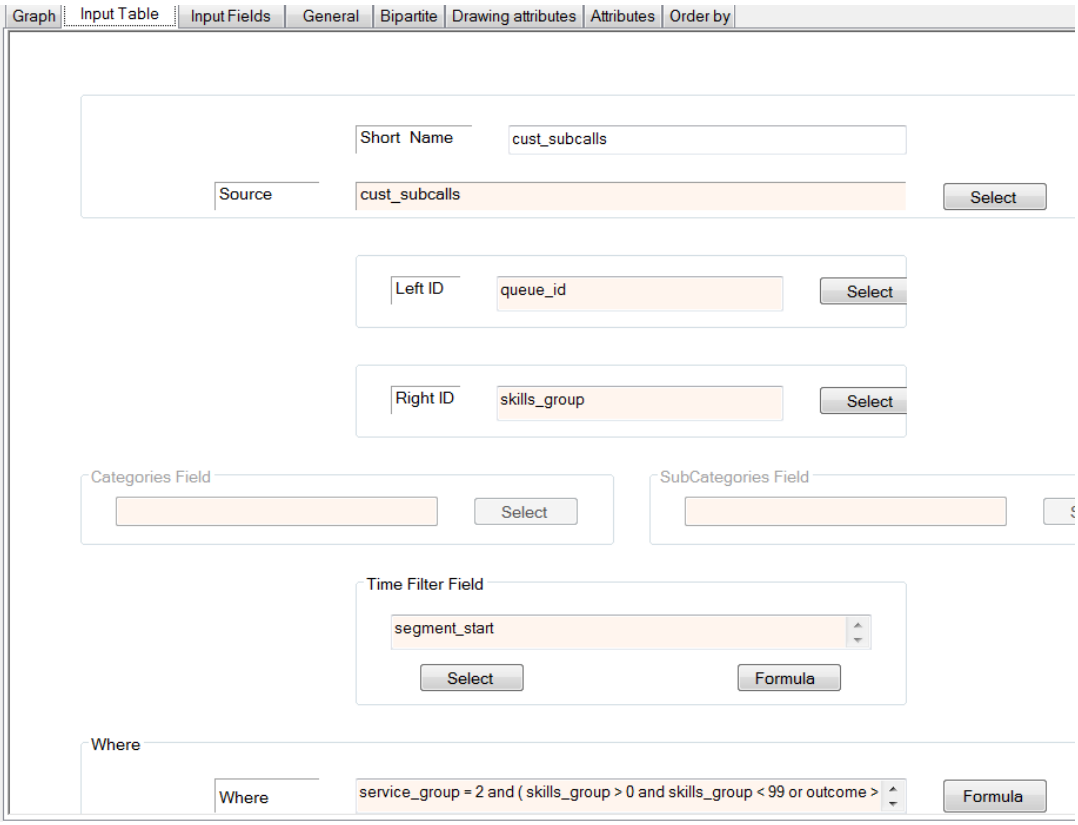


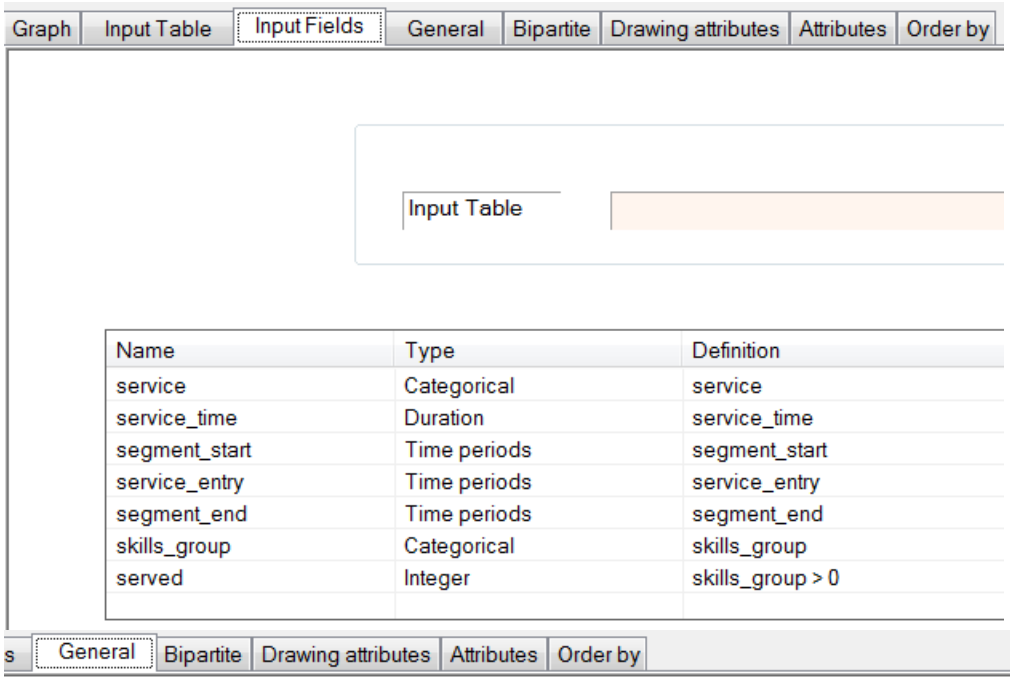


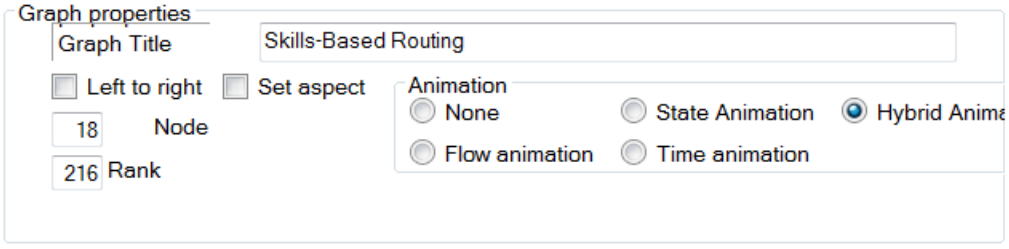
5. Hybrid animation: Process Mining + Operations Research view

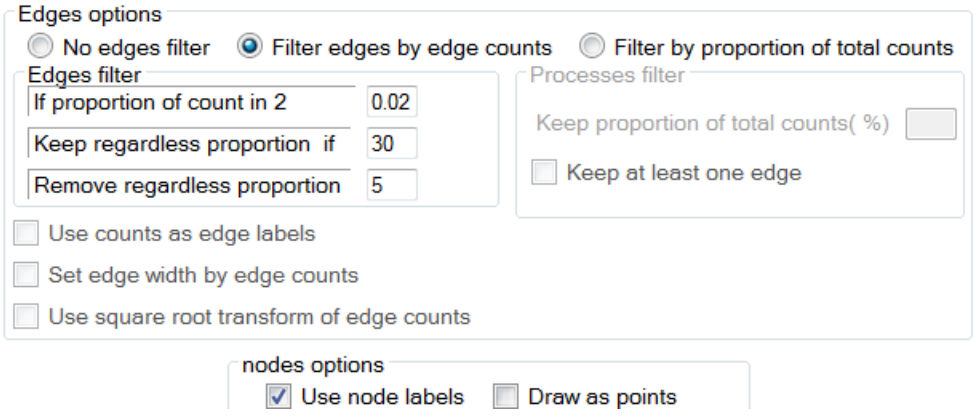


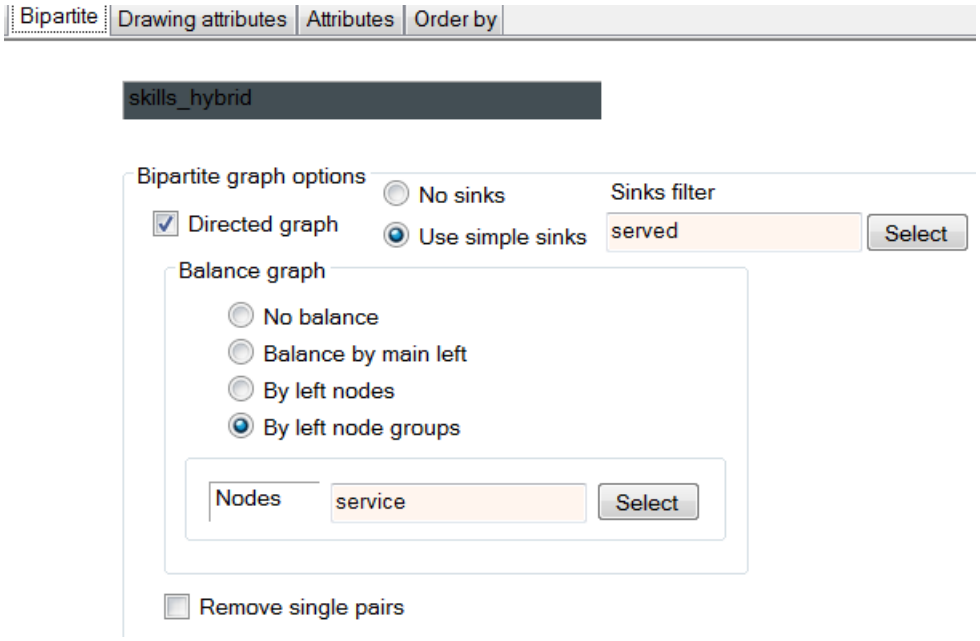

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Short Name
skills_hybrid
Graph Type
Flow Graph
Bipartite
Nodes definition
Node Dimension
1
2
Left ID
queue_id
Select
Right ID
skills_group
Graph Partiton Dimension
1
2
3
Categories
Name
Select
Filter
none
Formula
Subcategories
Name
Select
Filter
none
Formula
Time Filter
v > 07:30:00
Formula

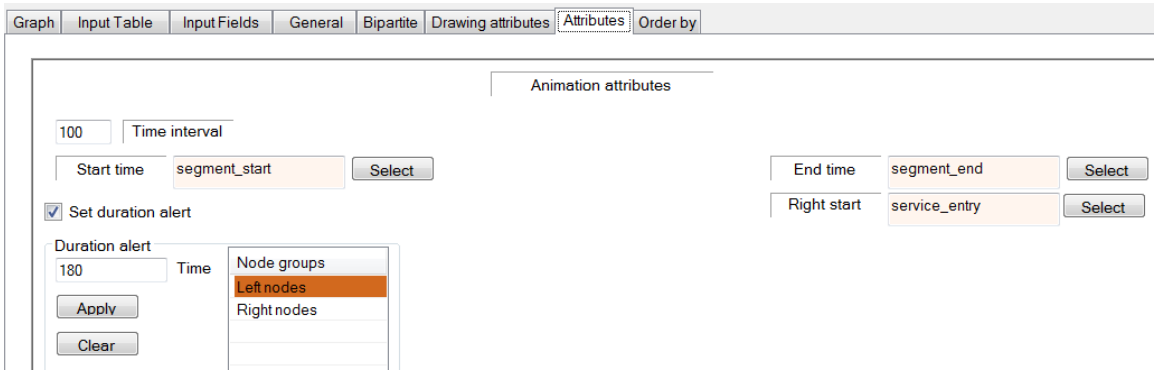

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Short Name
cust_subcalls
Source
cust_subcalls
Select
Left ID
queue_id
Select
Right ID
skills_group
Select
Categories Field
Select
SubCategories Field
Time Filter Field
segment_start
Select
Formula
Where
Where
service_group = 2 and (skills_group > 0 and skills_group < 99 or outcome >
Formula

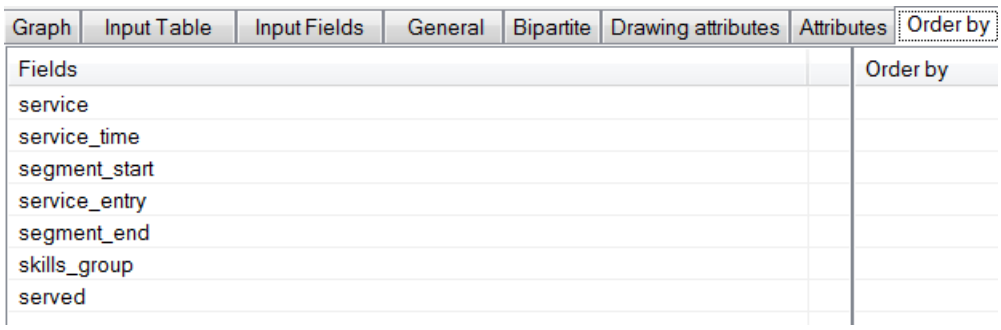

Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Input Table
Name
Type
Definition
service
Categorical
service
service_time
Duration
service_time
segment_start
Time periods
segment_start
service_entry
Time periods
service_entry
segment_end
Time periods
segment_end
skills_group
Categorical
skills_group
served
Integer
skills_group > 0
General
Bipartite
Drawing attributes
Attributes
Order by


Graph properties
Graph Title
Skills-Based Routing
Left to right
Set aspect
18
Node
216
Rank
Animation
None
State Animation
Flow animation
Time animation


Edges options
No edges filter
Filter edges by edge counts
Filter by proportion of total counts
Edges filter
If proportion of count in 2
0.02
Keep regardless proportion if
30
Remove regardless proportion
5
Processes filter
Keep proportion of total counts(%)
Keep at least one edge
Use counts as edge labels
Set edge width by edge counts
Use square root transform of edge counts
nodes options
Use node labels
Draw as points

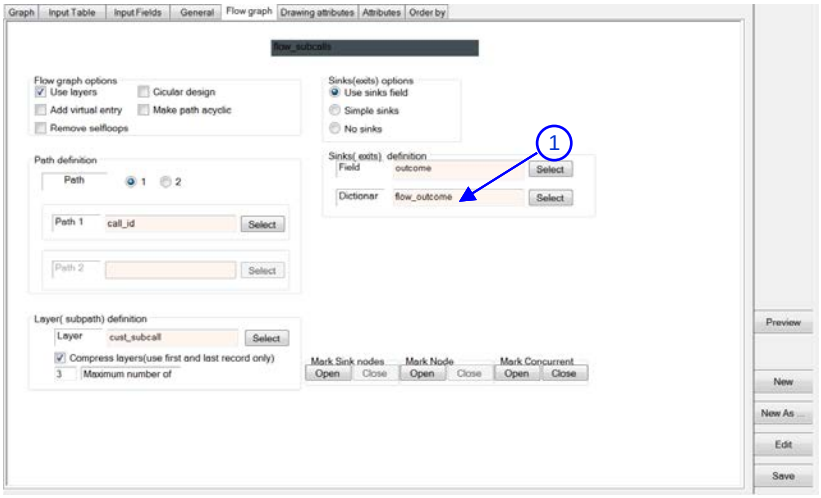

Bipartite
Drawing attributes
Attributes
Order by
skills_hybrid
Bipartite graph options
Directed graph
No sinks
Use simple sinks
Sinks filter
served
Select
Balance graph
No balance
Balance by main left
By left nodes
By left node groups
Nodes
service
Select
Remove single pairs


Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Animation attributes
100
Time interval
Start time
segment_start
Select
End time
segment_end
Select
Right start
service_entry
Select
Set duration alert
Duration alert
180
Time
Apply
Clear
Node groups
Left nodes
Right nodes


Graph
Input Table
Input Fields
General
Bipartite
Drawing attributes
Attributes
Order by
Fields
Order by
service
service_time
segment_start
service_entry
segment_end
skills_group
served

Exercise 2: Modify a graph – change sinks dictionary

1. Modify *flow_subcalls* graph (which was created in Part I): change sinks dictionary from *flow_outcome* to *outcome*.



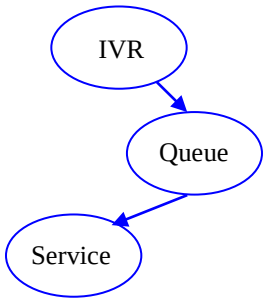
2. Process the edited flow_subcalls graph for April 2, 2001.
3. View graph via SEEGraph viewer.

Exercise 3: Create graph – use NEW AS option

1. Create *aggregated outcome* dictionary (see Appendix 2)
2. Create new graph by using *New AS* option from *flow_subcalls* graph, use *aggregated outcome* dictionary as sinks dictionary. Group sinks into Normal, Abnormal and Transfer sinks groups.

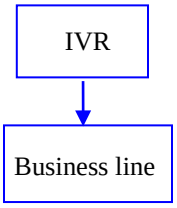
Exercise 4: Create Calls Flow graph

Create calls flow graph with following structure:



(a) graph with layers, with maximum of 5 layers.
(b) graph without sinks.
(c) graph with 2 node dimensions.

For this task you will need to build a few procedures based on the source table *cust_subcalls* and create regular study dictionary.
The current structure of *cust_subcalls* table is:



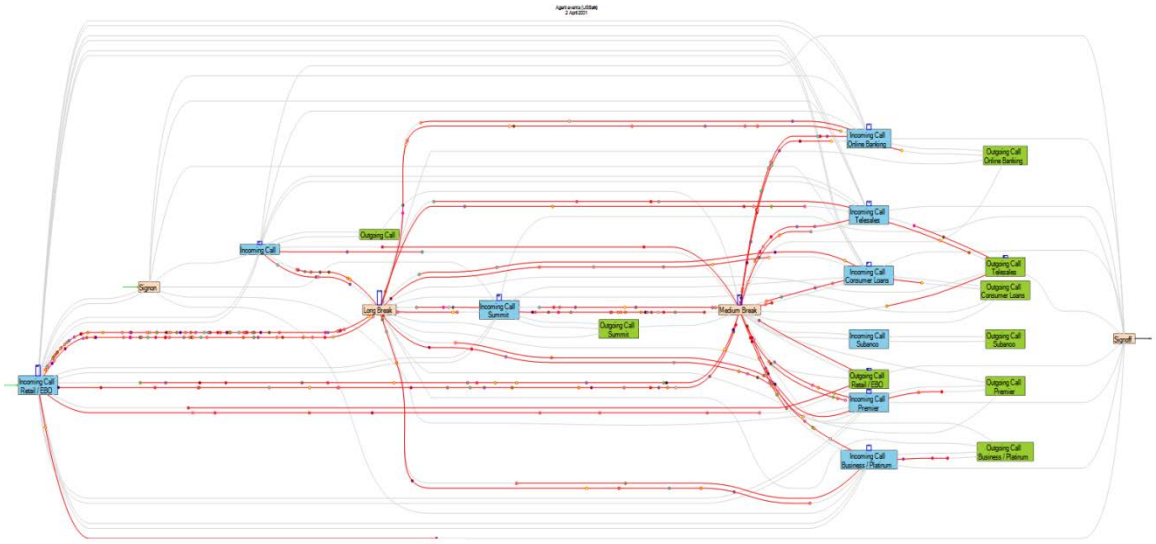
See Appendix 4 for a description of *cust_subcalls* table.
See Appendix 2 for a description how to create dictionary.
See Appendix 6 for a description how change type of field.

Exercise 5: Create Agent events graph

Create the following agent events graph by using *USBank* data:

- hybrid animation type
- with simple sinks
- 2 node dimension
- source table agent_events.

See Appendix 4 for a description of *agent_events* table.

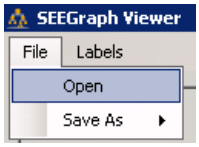


Part III SEEGraph View: User Guide

SEE Graphs Animations

Opening an animation file:

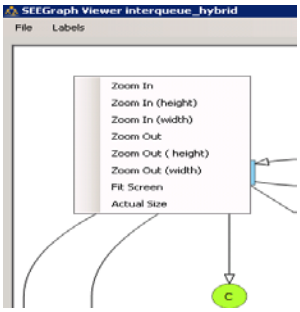
Click ‘File -> Open’ and choose the desired animation from the available animation list:



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

Zooming:

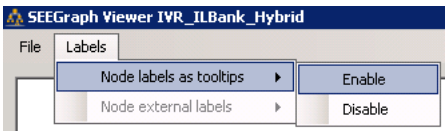
Right clicking the screen will open the following dialog box:



Choose the desired option (e.g. Zoom-in, Zoom-in height).

Labels:

In order to see the node labels, select ‘Labels -> Node labels as tooltips -> Enable’:

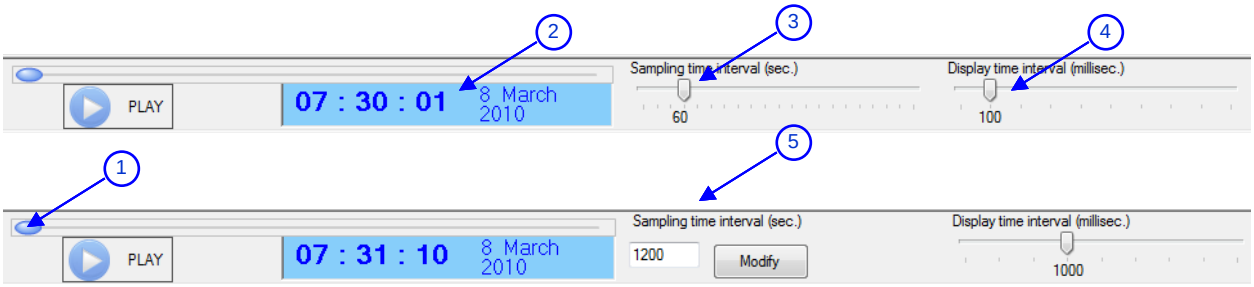


Sampling time interval:

In the animation files that include the word Network (structural animations), each frame in is based on an aggregation of all the events that occurred during a specific time interval. In the Process-Mining (flow) animations, the OR (queuing perspective) animations and the Hybrid animations each frame represents system state by the end of a specific time interval. For all cases, the time-interval lengths can be modified using the 'Sampling time interval' adjustment bar.

Display time interval

This adjustment bar controls the time between displays of consequent animation frames. Increased (decreased) 'Display time interval' value makes the animation run slower (faster).



(1) Time scroll

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

(3) Sampling time interval (sec.) 60 seconds means that if current time is 7:31:10 then the next picture will be at 7:32:10.

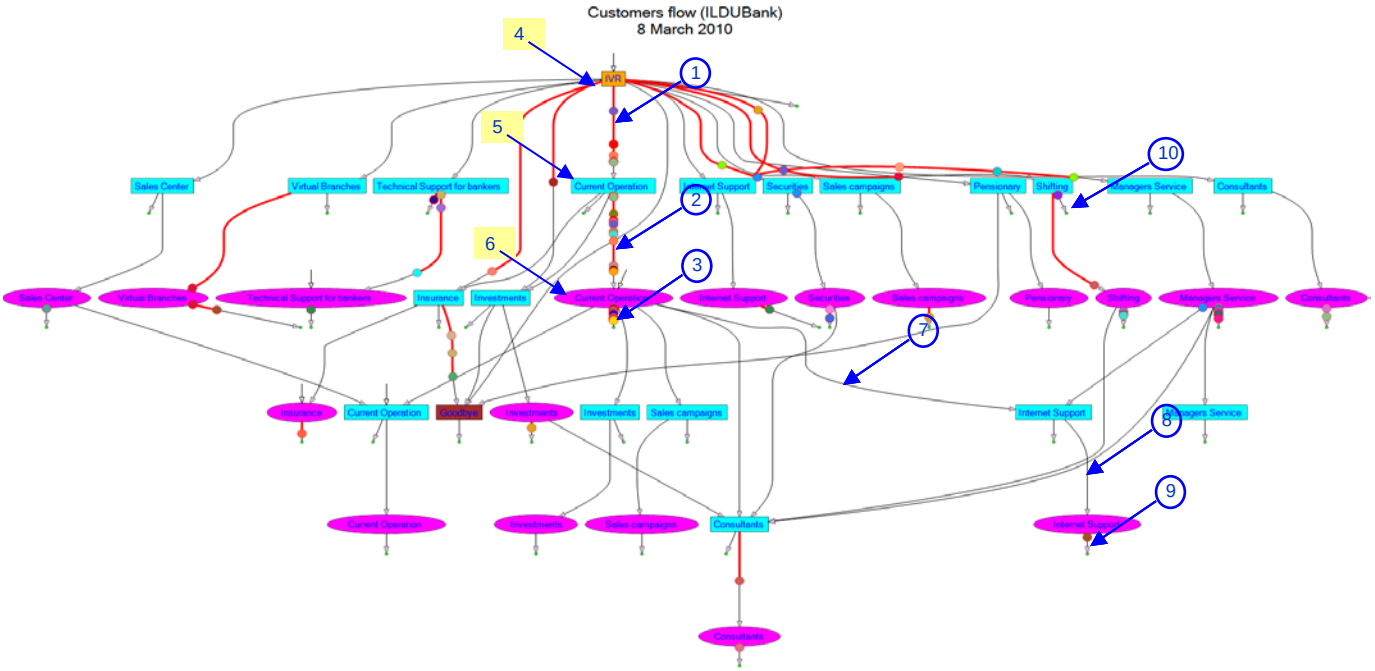
(4) Display time interval (milisec.) time between consequent updates of the picture.

(5) Sampling time interval (sec.) for time animation – 1200 seconds.

Examples

At your convenience we provide four basic animation screenshots, to further explain the different views available in SEEGraphView:

1. The Process Mining (Flow) view:

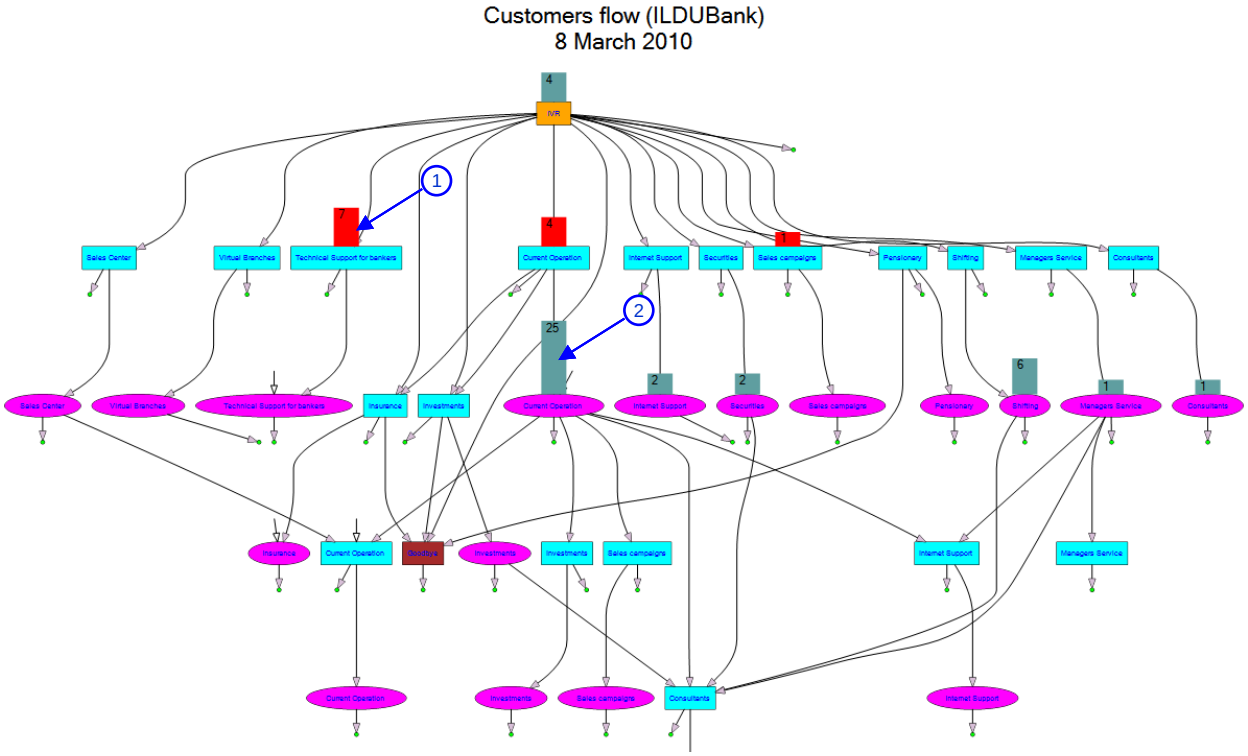


- queue start event
- service start event

Arcs – correspond to time spent in queue/service/IVR

(1) On arc: customer in the IVR. Duration on arc corresponds to IVR time.
(2) On arc: customer in queue. Duration on arc corresponds to WAIT time.
(3) On arc: customer in service. Duration on arc corresponds to SERVICE time.
(4) Node: corresponds to IVR start time (start event).
(5) Node: corresponds to Queue entry of Current Operations customer (entry event); this is the first customer sub-call
(6) Node: corresponds to Service entry of Current Operations customer (entry event); this is still the first customer sub-call
(7) On arc: SERVICE time for a Current Operations customer; at the end of the service he is transferred to the Internet support queue; this is still part of the first sub-call
(8) On arc: Waiting time in the Internet Support queue for customers who were transferred from a different service; this is the second sub call.
(9) On arc: Service time in the Internet Support queue for customers who were transferred from a different service; this is the second sub call
(10) On arc: customers who abandon. Duration on arc corresponds to WAIT time before abandonment.

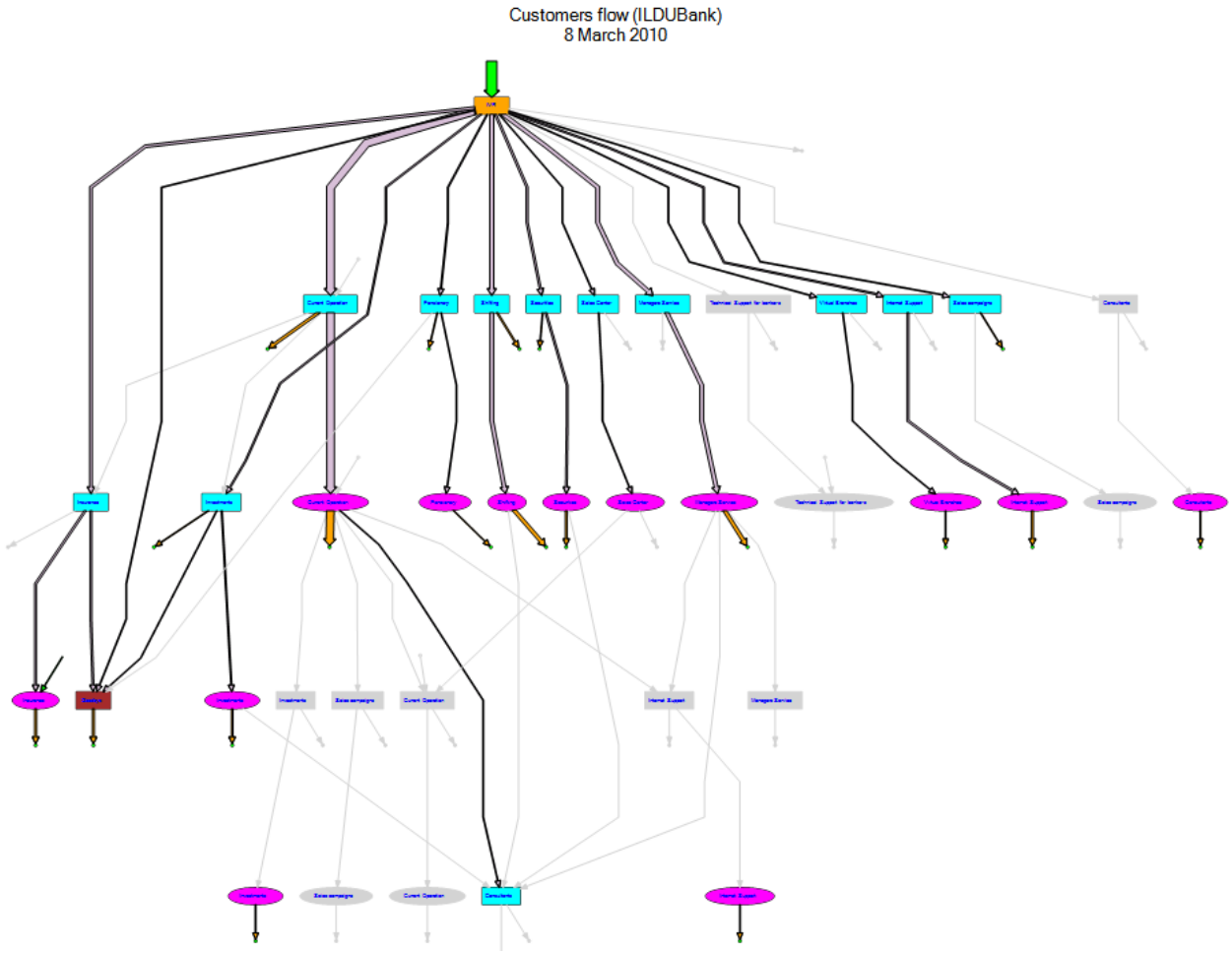
2. The Operations Research (or queueing) view:



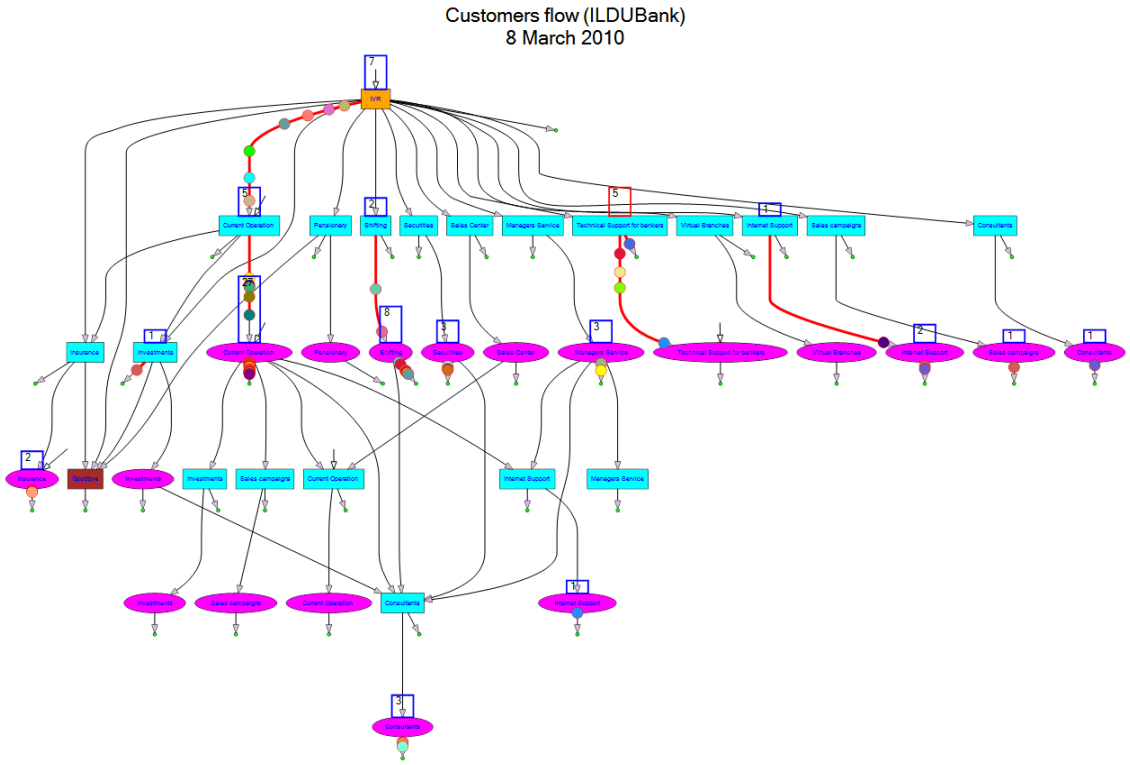
(1) Number of customers in the Technical Support queue. The red color indicates that maximal waiting time is above 120 seconds.
(2) Number of Current Operations customers that receive service.

Calculation: count events between start time and end time

3. The network (structure) view:



4. The hybrid (Process Mining + Operations Research) view:

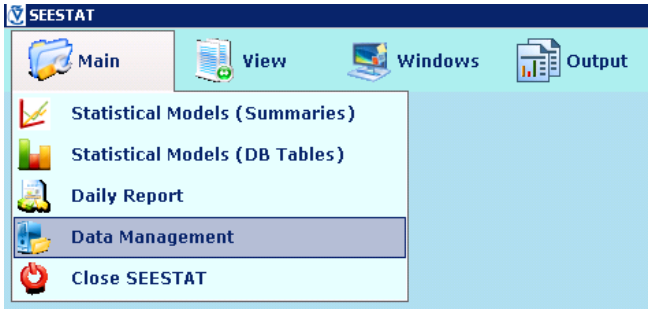


Appendix 1 Creation of SEEStat user extension

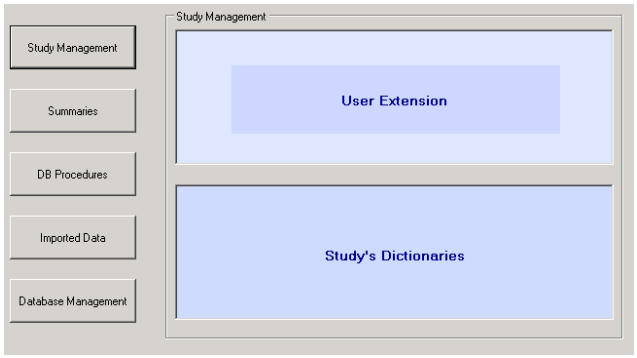
Do it only one time, there is no need to repeat it afterwards. Note: user capacity in the SEE Server is limited to 150MB.

1.1 Register user folder

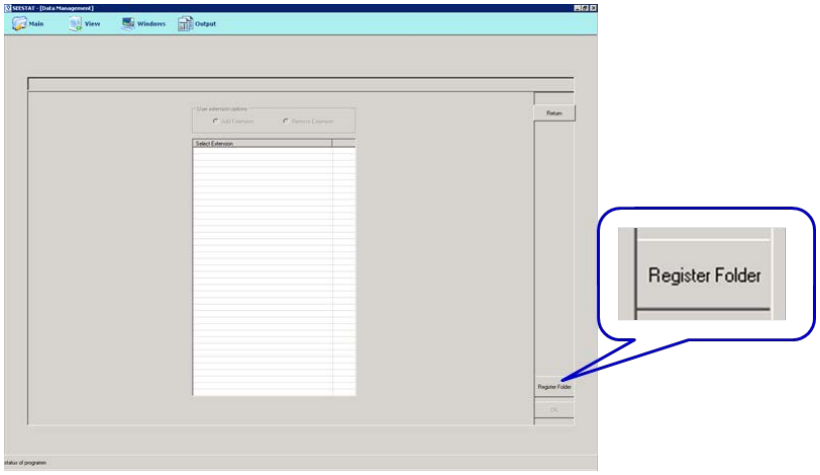
Click Main –> Data Management



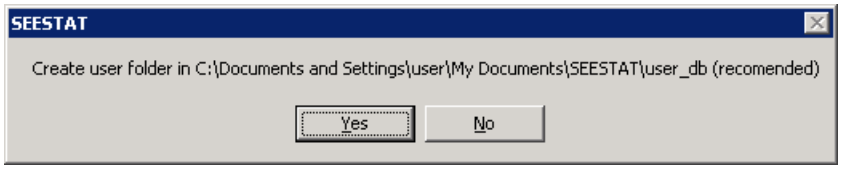
Click Study Management –> User Extension



Click Register Folder

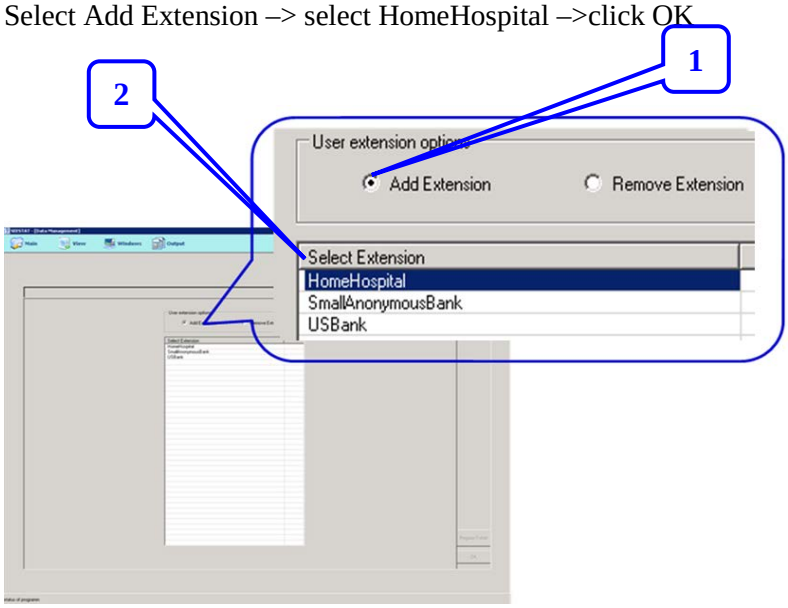


Click Yes



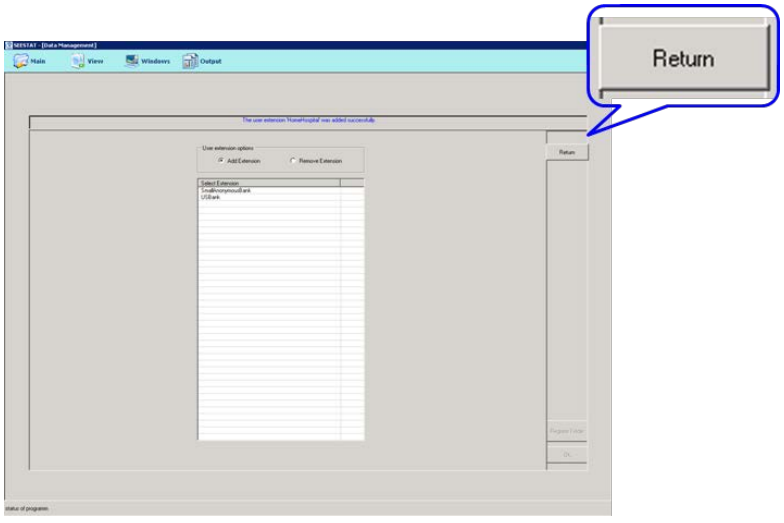
1.2 Add user extension

Select Add Extension –> select HomeHospital –>click OK



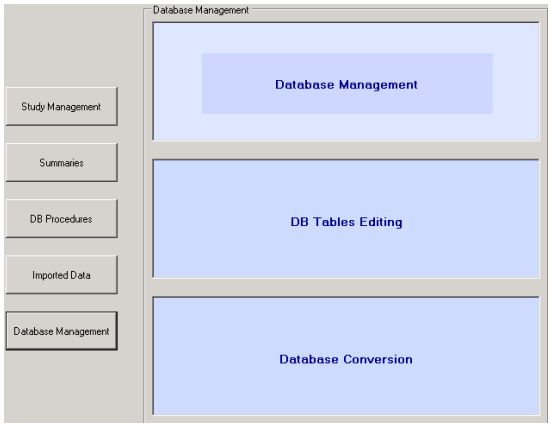
If the message: "*Unable to add or rename extension of the open study HomeHospital*" appeared, reopen SEEStat. Click Main –> Data Management –> Study Management –> User Extension. Select Add Extension –> select HomeHospital –>click OK

If the above message did not appear, continue with next step:
Click Return



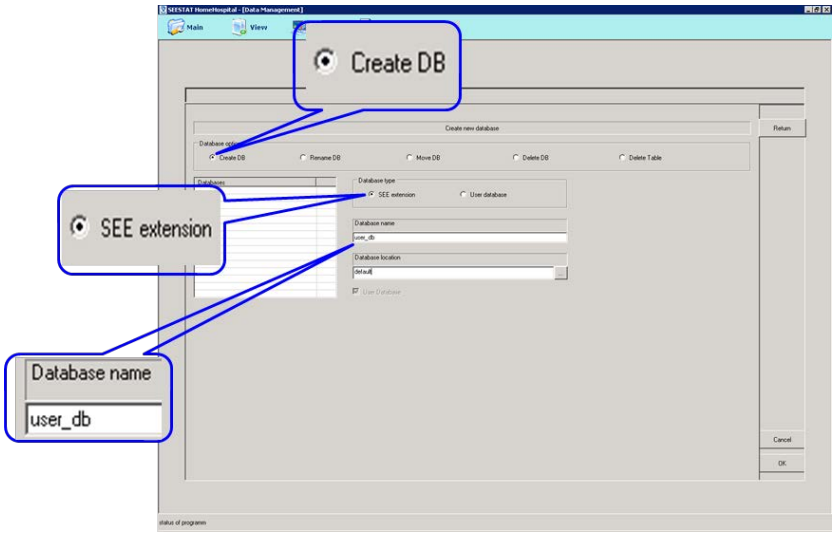
1.3 Create user database

Select Database Management –> Database Management



Select HomeHospital study and OK.

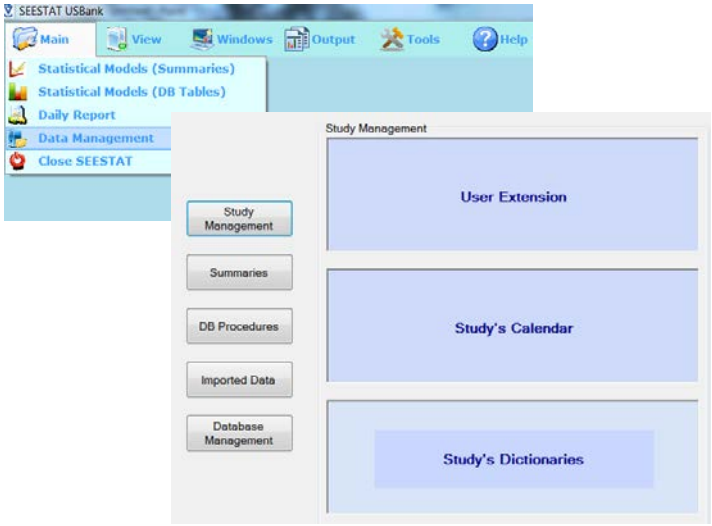
From database options select Create DB (radio-button), select database type: SEE extension (radio-button),
Under database name fill: user_db, click OK



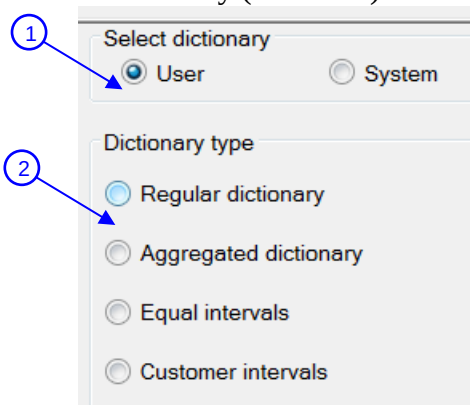
Appendix 2 Create Study's Dictionaries

Regular dictionary

Click *Main->Data Management->Study Management-> Study's Dictionaries*



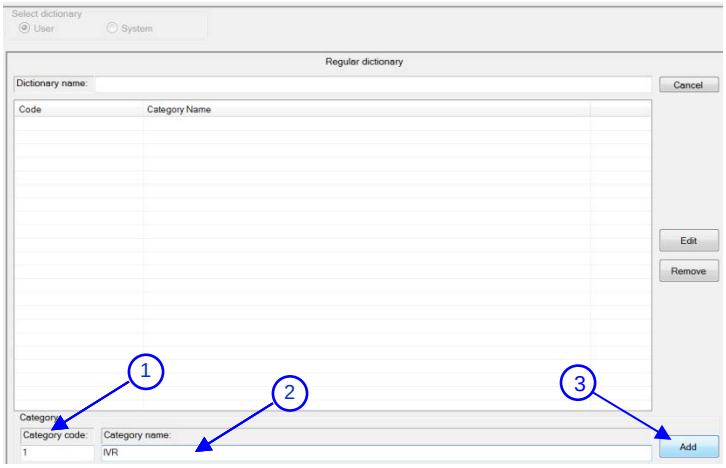
Select study (if needed).



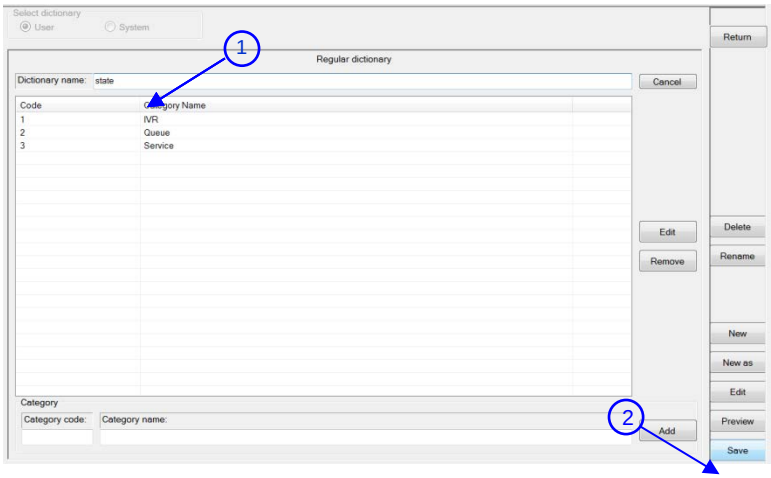
(1) Select dictionary: *User*
(2) Select dictionary type: *Regular dictionary*

For example needs to create dictionary this following structure:

code	name
1	IVR
2	Queue
3	Service



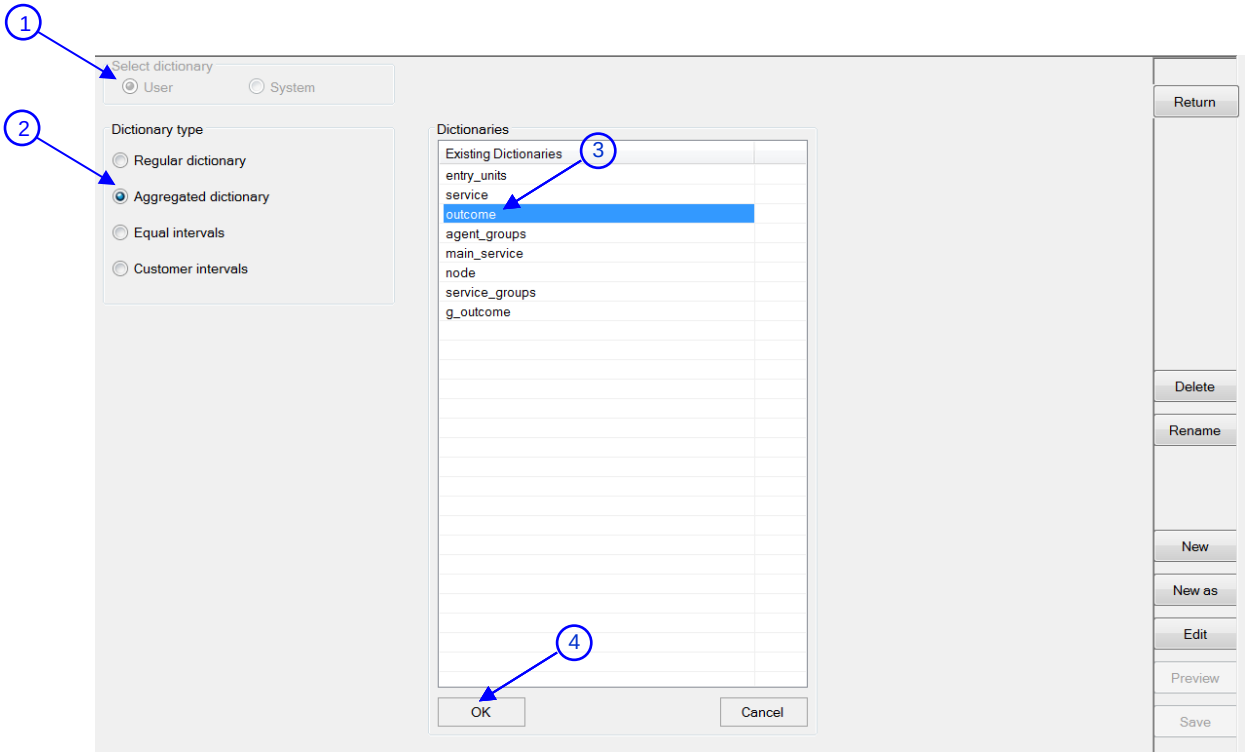
(1) Fill category code: *1*.
(2) Fill category name*: IVR*.
(3) Click *Add* button. Fill additional codes and names in the same manner.



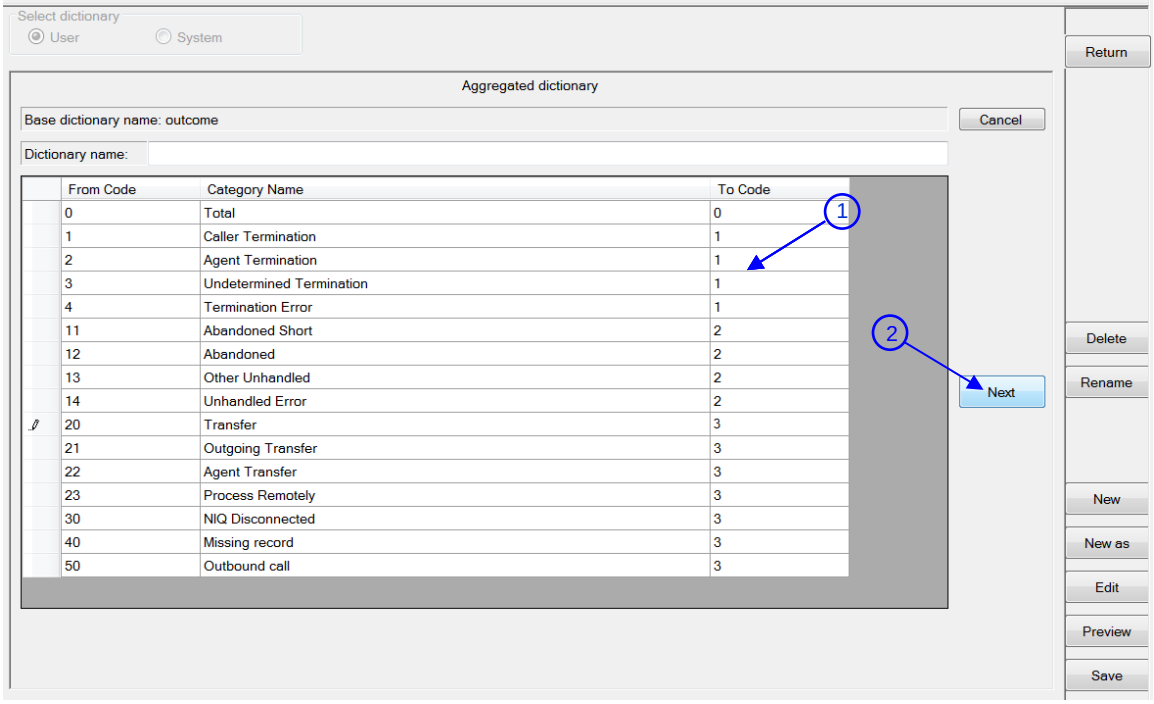
(1) Fill dictionary name.
(2) Click *Save* button.

Aggregated dictionary

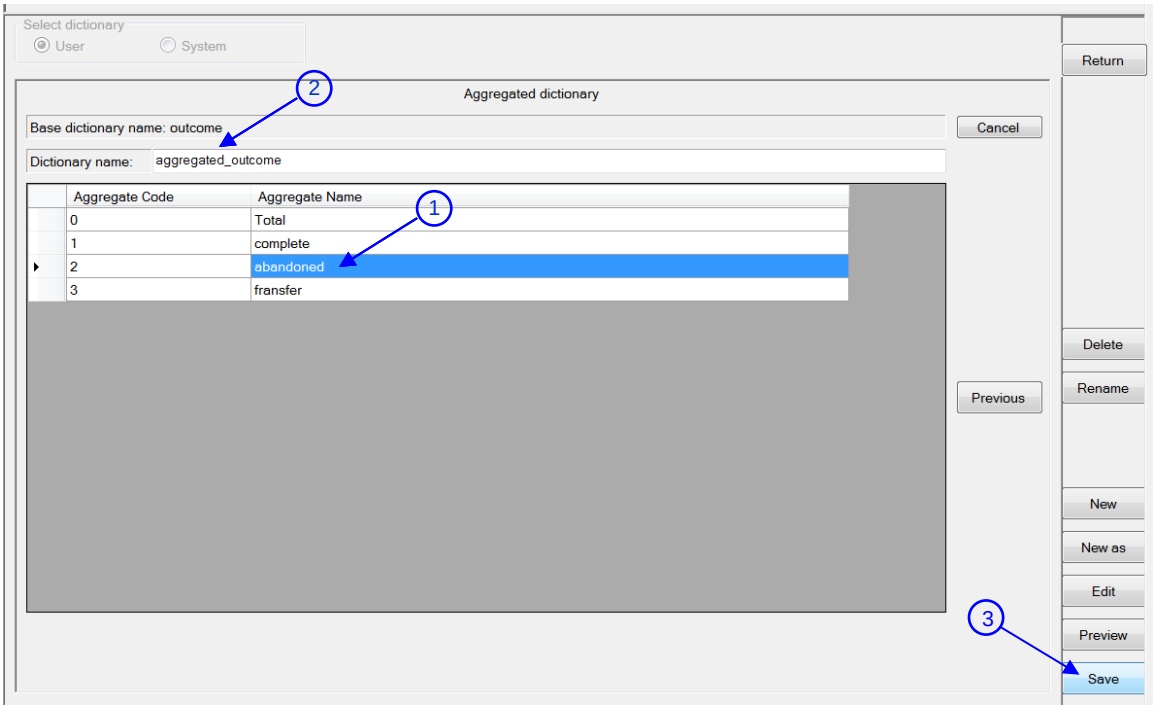
Click *Main->Data Management->Study Management-> Study's Dictionaries*



(1) Select dictionary-> *User*
(2) Select dictionary type-> *Aggregated dictionary*
(3) Select existing dictionary: *outcome*
(3) Click *OK*

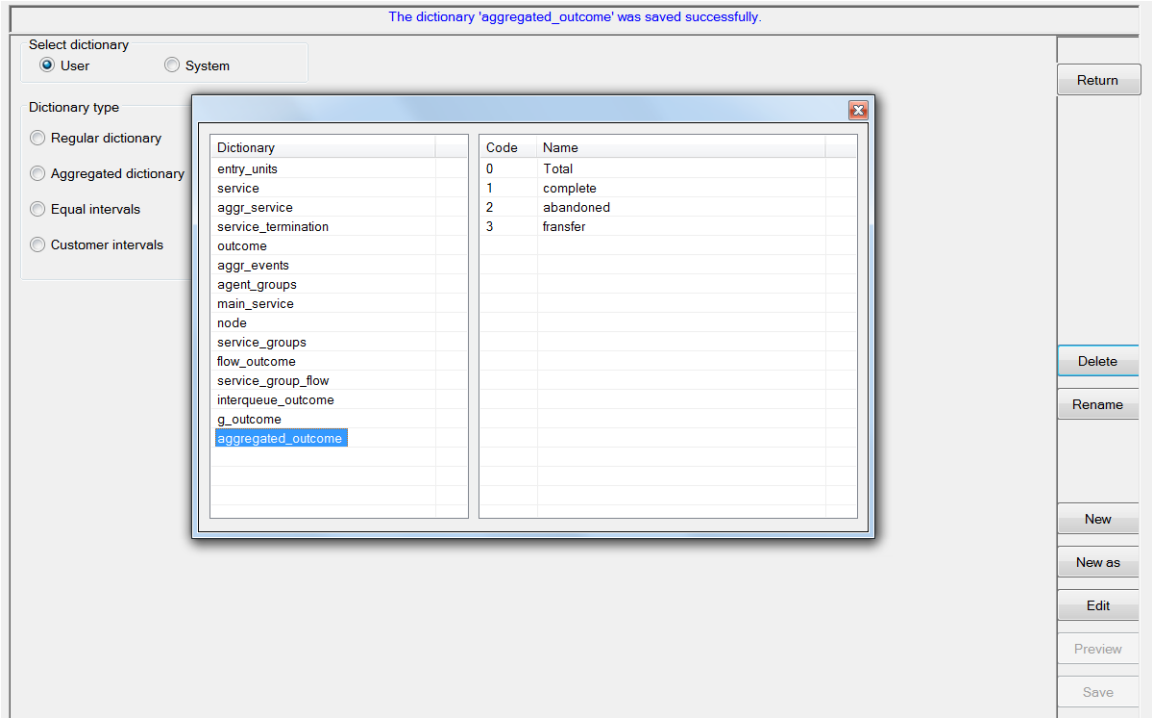


(1) Fill *To Code* values
(2) Click *Next* button



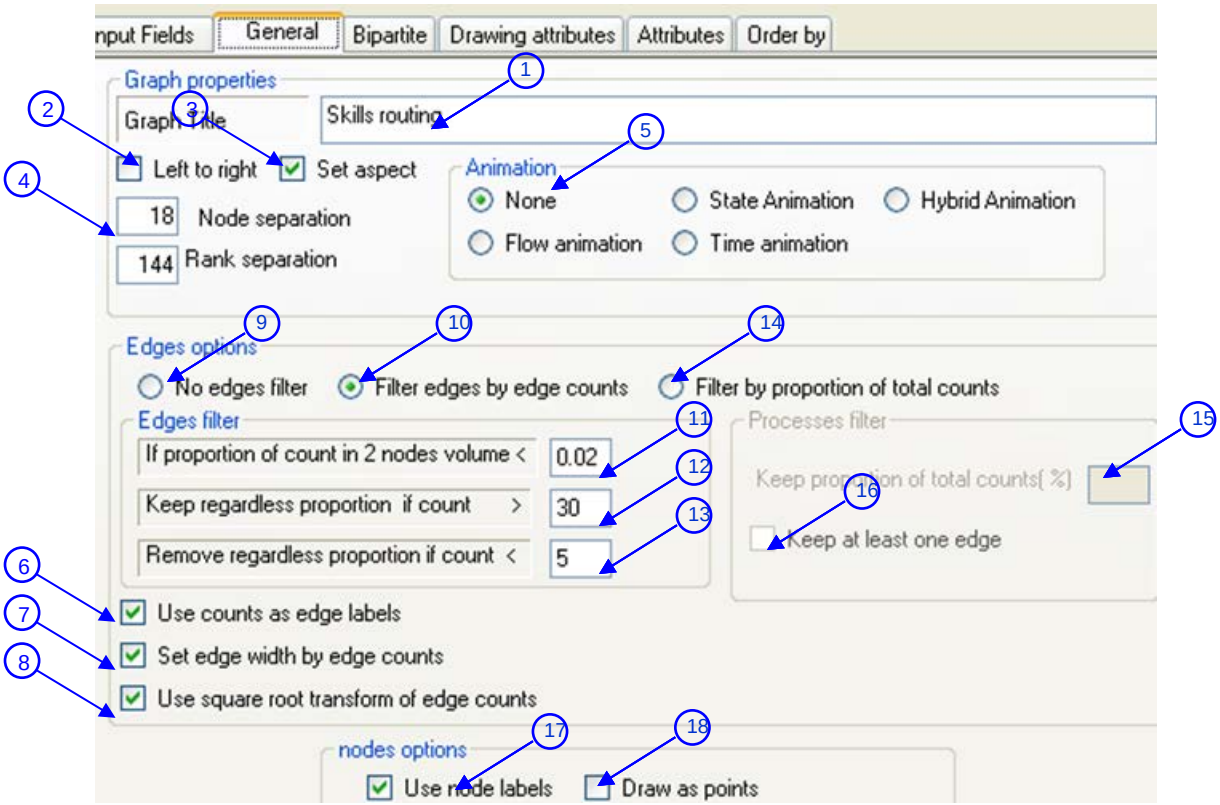
(1) Change *aggregated names*
(2) Fill *dictionary name*
(3) Click *Save*

Click View->*Dictionaries->aggregated_outcome*



Appendix 3 SEEGraph Selected Tabs

The General tab



Graph properties:

(1) **Graph Title:** This option allows inserting a meaningful name to the graph, which will then be presented on the top of the screen. Note that the name of the study and the date are placed automatically as part of the title (no need to write those).
(2) **Left to right:** The option (when selected) places the graph flow from left-to-right instead of top-down.
(3) **Set aspect**: set full screen graph view (*not active for current version*)
(4) **Node separation**:
Rank separation:

(5) Select animation type: **None**: without animation; Flow animation depicts customer dynamics as they flow through the states of the graph; State animation depicts the queueing perspective, i.e. presenting counts of customers in queue/in service; Time animation presents the intensity of the flow over time, with width of arcs corresponding to flow intensity. Lastly, the hybrid animation presents both the State view and the Flow view together.

Edge options:

(6) **Use counts of edge as labels**: Presents the number of flowing cases through a certain edge (note that it is not recommended to use both width and count).
(7) **Set edge width by edge counts**: Setting the width of the arcs correspondingly to the number of customers that flow through the specific arc.
(8) **Use square root transform of edge counts**: An option that scales widths to be non-linear (square root function) of the counts.
(9) **No edges filter**: Every possible route between two nodes is presented.
(10) **Filter edges by edge counts**: Filtering some of the routes from the process map.

Filter options:

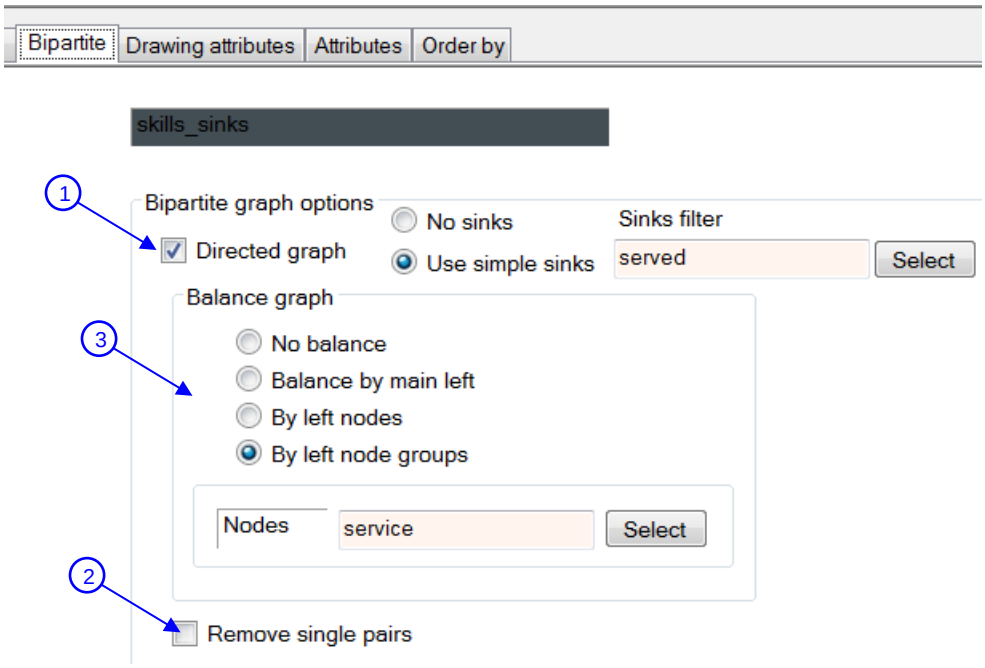
(11) **If proportion of count in 2 nodes volume < (e.g.** *0.02)*: If the proportion of traffic through an arc is smaller than x% of the total flow in pairs of nodes (e.g. 2%) then it will be eliminated from the graph.
(12) **Keep regardless proportion if count >** *30*: This option keeps the arc (even if it is below x% of total count) if it is "important enough", i.e. with value being over some y (e.g. y= 30 in our example).
(13) **Remove regardless proportion if count <** *5*: The option removes edges that are above x%, if they are below a certain absolute count z (e.g. z=5 in our example). So, if an edge qualifies proportion-wise it might still be removed due to total count constraints.
(14) **Filter by proportion of total counts**: Options that consider the total flow (and not only pairs of nodes).

(15) **Keep proportion of total counts (%)**: Keeps a proportion of total counts per arc.
(16) **Keep at least one edge**: Keeps at least one edge, if all edges qualify for removal.

Nodes options:

(17) **Use node labels:** Nodes can be empty, so checking this option would present the node label (e.g. name of activity).
(18) **Draw as point:** This option allows drawing nodes as points (takes zero space).

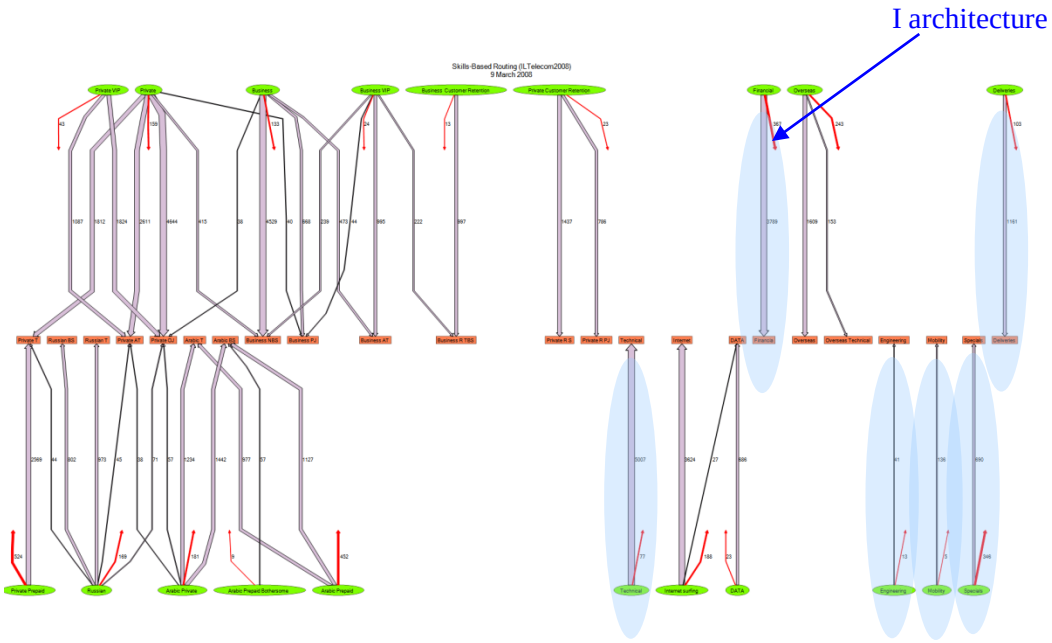
The Bipartite tab



Bipartite graph options:

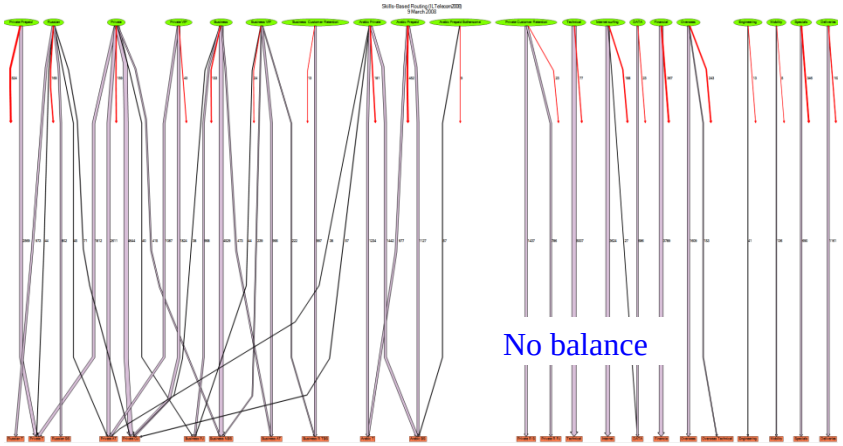
(1) Direct graph – arcs are directed (in contrast to an undirected graph, where arcs have no direction and traffic can go both ways)

(2) Remove single pairs – remove one-to-one connections, i.e. I-architectures.

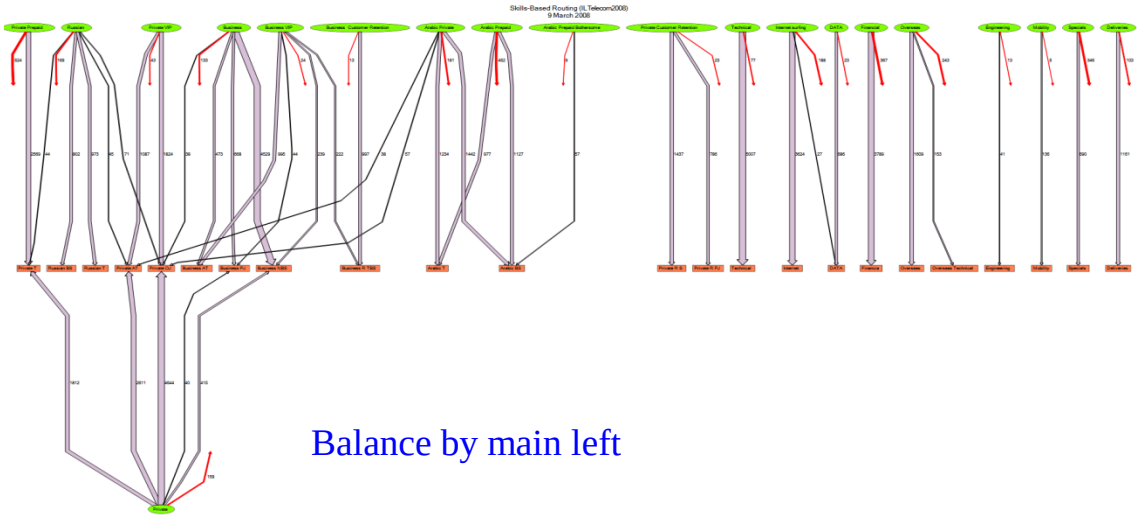

I architecture
Skills-Based Routing (ILTelecom2008)
9 March 2008

(3) Balance graph options:

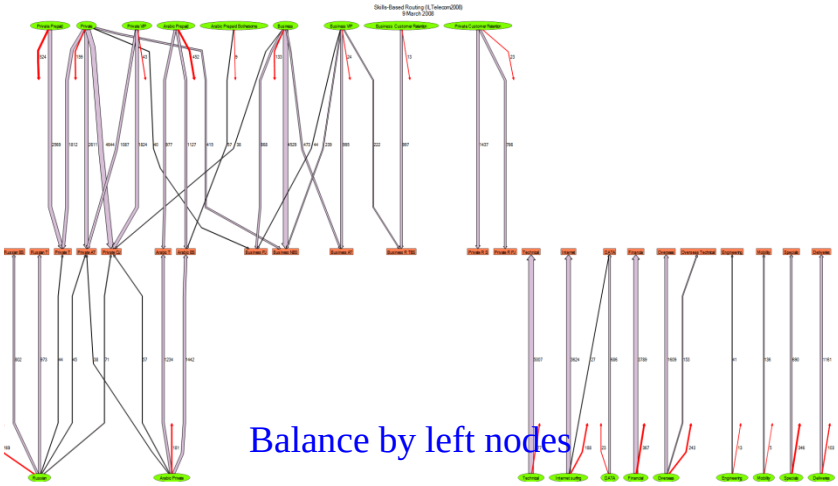
- **No balance** – One set (queues) on the one side, the other set (service pools) on the other side.



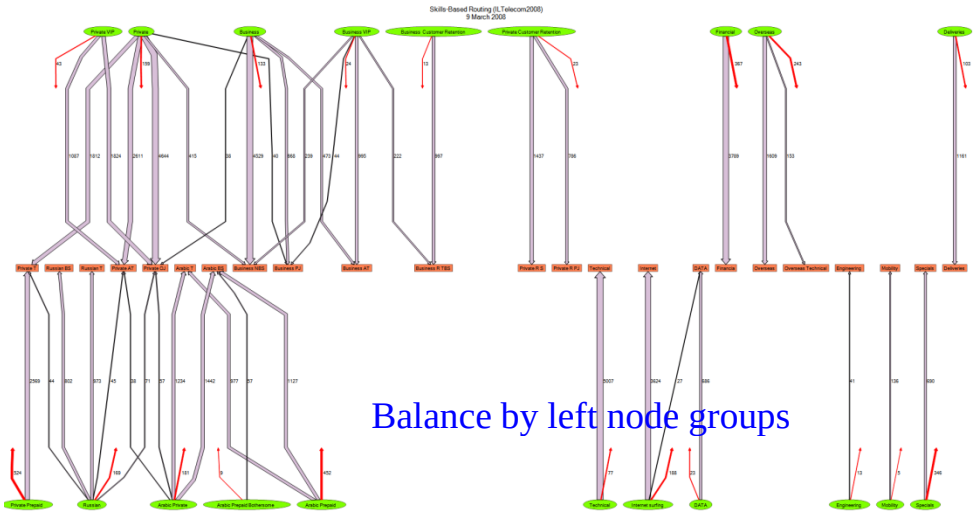
- **Balance by main left** – Most connected components move below



- **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



Appendix 4 Selected USBank tables

The agent_events table

1. agent – agent extension number.
2. primary_service – service the agent is skilled to provide (see following *service* dictionary).
3. event_start – time in seconds at which the segment is started.
4. event_end – end_time – time in seconds at which the segment is ended.
5. start_time – date/time at which the segment is started.
6. end_time – date/time at which the segment is ended.
7. duration – amount of time agent performing an event specified in field event_id.
8. event_id – event codes for idle states (40–49), breaks (60–62), available state (50), sign–on states (20–21), sign–off states (30–31), agent originated (2) or agent answered (1) call segment (see following *agent events* dictionary).
9. business_line – associated call received at least one service – 1, or otherwise – 0.
10. service – type of service received by the caller (see following *service* dictionary).
11. node – identifier of the site where the agent is situated.
12. record_id – ID number assigned to the record, this is created for all the segments of a particular day.
13. agent_group – skill–group is defined to be a group of agents that have the same skill–set to serve the different service types.
14. main_service – main–service is defined to be the most important service type that a skill–group serves. More specifically, the main–service is defined according to the percentage of the agent calls from each service type and the percentage of the service type calls in each agent group.

The customer sub–calls table

1. call_id – universal identifier associated with the entire call.
2. cust_subcall – sequence number of service that a caller received during his call.
3. record_id – ID number assigned to the record, and is created uniquely for all segments of particular day.
4. node – identifier of the site where the call is currently being processed.
5. service_group – service group that handled the call (see following *service group* dictionary).
6. service – type of service received by the caller (see following *service* dictionary).
7. first_service – first type of service requested by the caller from the primary agent (see following *service* dictionary).
8. segment_start – time in seconds at which the segment is started.
9. queue_exit – time in seconds at which the call exits the queue.
10. service_entry – time in seconds at which the call enters the agent.
11. segment_end – time in seconds at which the segment ends.

12. seg_type – state of the call (Begin/End/Intermediate) (see following *segment type* dictionary).
13. outcome – cause of call termination (Handled/Transferred/Abandoned/..) (see following *outcome* dictionary).
14. seg_parties – type of resource that answered the call (Agent/Supervisor/Conference/..) (see following *segment parties* dictionary).
15. wait_time – delay time plus queue time.
16. queue_time – queue time.
17. preservice_wait – ring time and call_type time.
18. service_time – talk time and hold time.
19. hold_time – amount of time a caller spent on hold on an agent's teleset.
20. undefined_time
21. party_answered – resource/code number that answered the call; for example, if the number is greater than 10000, then an agent answered the call.
22. agent_group – skill–group is defined to be a group of agents that have the same skill–set to serve the different service types.
23. main_service – main–service is defined to be the most important service type that a skill–group serves. More specifically, the main–service is defined according to the percentage of the agent calls from each service type and the percentage of the service type calls in each agent group.

Selected USBank dictionaries:

service dictionary	
1	Retail
2	Premier
3	Business
4	Platinum
5	Consumer Loans
6	Online Banking
7	EBO
8	Telesales
9	Subanco
10	Case Quality
11	Priority Service
12	AST
13	CCO
14	Summit
15	Quick&Reilly
16	Mortgage
17	BPS

service group dictionary	
1	VRU
2	Business Line
3	Announcement
4	Message
5	NonBusiness Line
6	NonCC Service
8	Overnight Closed
9	Trunk
10	Incoming NonBusiness
11	Internal
12	Outgoing
15	Disconnected
99	Unknown

segment type dictionary	
1	customer call start
2	customer call start and end
3	customer call end
4	customer call middle segment
5	processed in another queue
6	outgoing
7	transfer
8	external transfer
9	agent to agent
10	supervisor key pressed
11	message key pressed
12	predictive message

outcome dictionary	
1	Caller Termination
2	Agent Termination
3	Undetermined Termination
4	Termination Error
11	Abandoned Short
12	Abandoned
13	Other Unhandled
14	Unhandled Error
20	Transfer
21	Outgoing Transfer
22	Agent Transfer
23	Process Remotely
30	NIQ Disconnected
40	Missing record
50	Outbound call

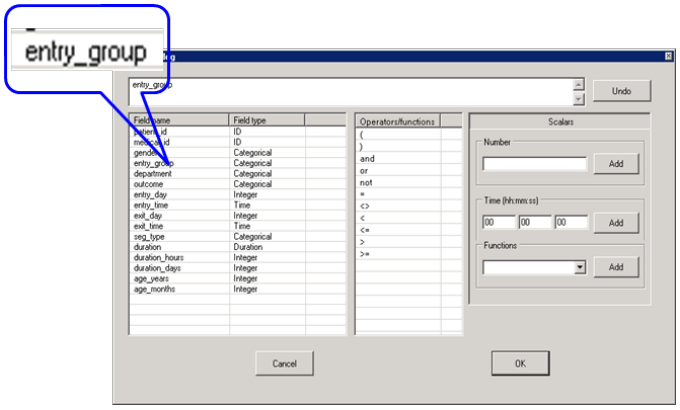
agent events dictionary	
1	Incoming Call
2	Outgoing Call
20	Signon
21	Internal Signon
30	Signoff
31	Internal Signoff
40	Idle Noreason
41	Idle Break
42	Idle Papers
43	Idle Back to Customer
49	Idle Signon
50	Available
60	Short Break
61	Medium Break
62	Long Break

segment parties dictionary	
10	Trunk
11	Trunk + Conference
12	Trunk + Emergency
13	Trunk + Conference + Emergency
20	Agent
21	Agent + Conference
22	Agent + Emergency
23	Agent + Conference + Emergency
30	Announcement
31	Announcement + Conference
32	Announcement+ Emergency
33	Announcement + Conference + Emergency
40	Voice port
41	Voice port + Conference
42	Voice port + Emergency
43	Voice port + Conference + Emergency
50	Agent
51	Agent + Conference
52	Agent + Emergency
53	Agent + Conference + Emergency
80	Virtual trunk
81	Virtual trunk + Conference
82	Virtual trunk + Emergency
83	Virtual trunk + Conference+ Emergency
90	Interflow trunk
91	Interflow trunk + Conference
92	Interflow trunk + Emergency
93	Interflow trunk + Conference + Emergency

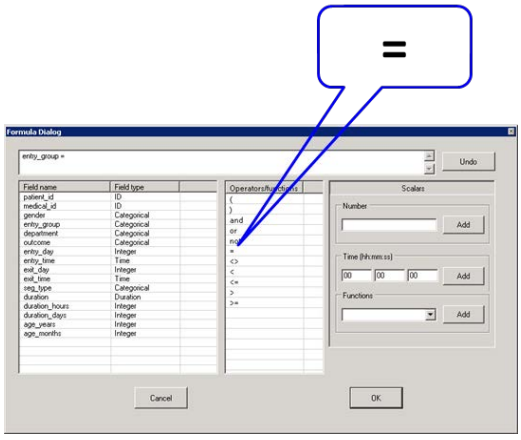
Appendix 5 How to type a formula in SEEStat

5.1 Fill in formula: *entry_group = 1*

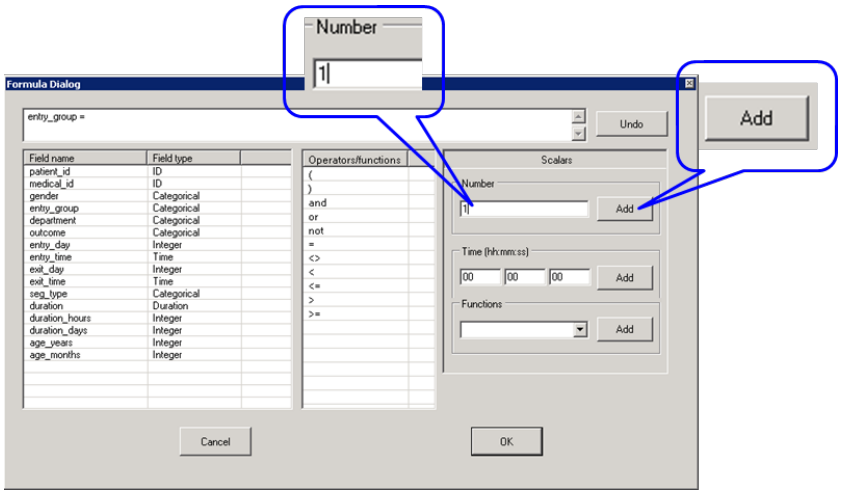
Select entry_group field in Field name list box.

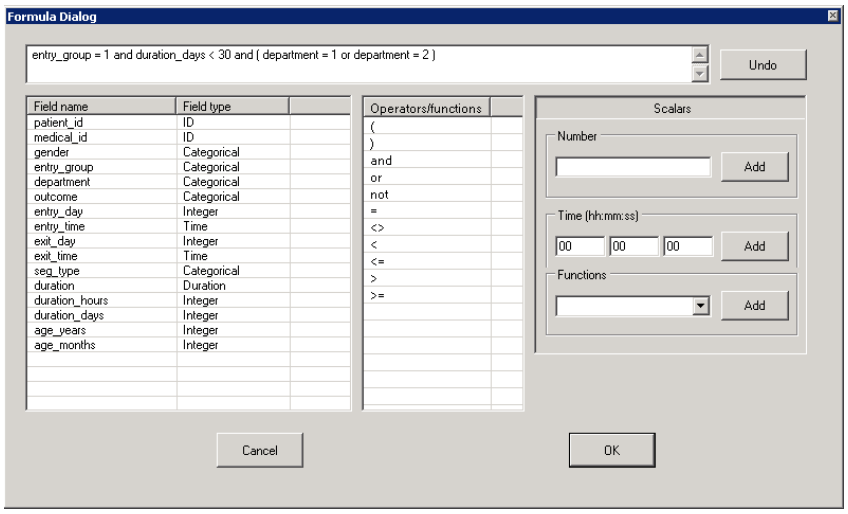


Click on operator *equal* (=) in Operators/functions list box.



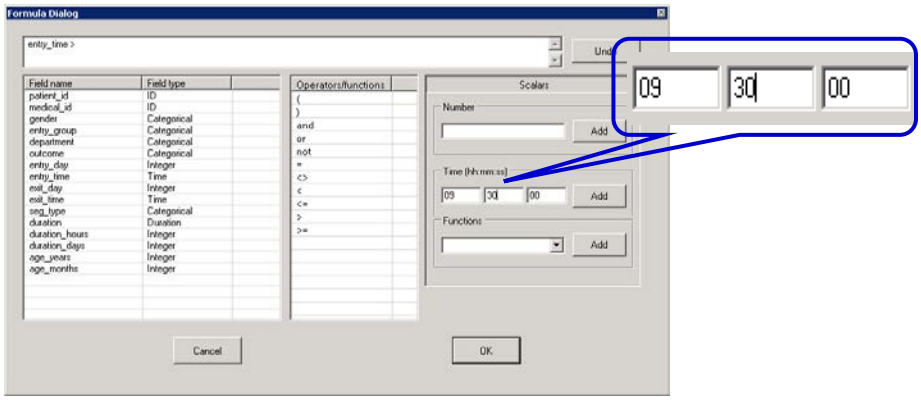
Write number 1 in text box in frame *Scalars* and click button Add.





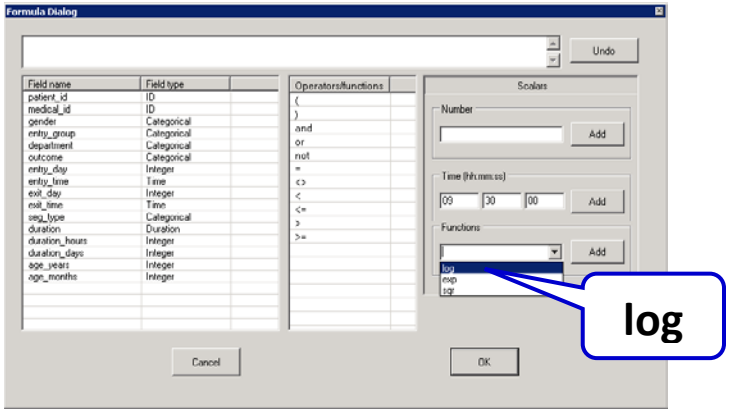
5.2 Fill in formula: *entry_time > 09:30:00*

Select *entry_time* field in Field name list box. Click on operator *more* (>) in Operators/functions list box. Fill in time *09:30:00* in text box in frame *Time (hh:mm:ss)* and click button Add.



5.3 Fill in formula: *log (duration)*

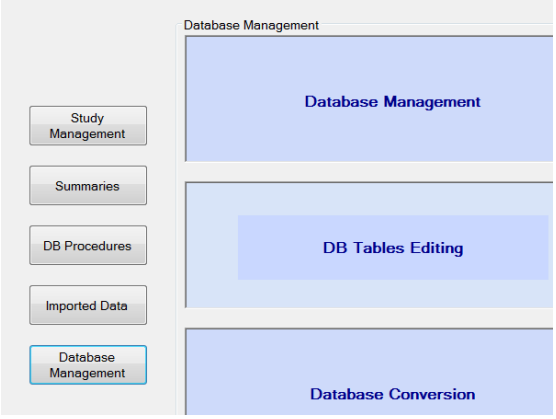
Select *log* in combo box in frame *Functions* and click button Add.



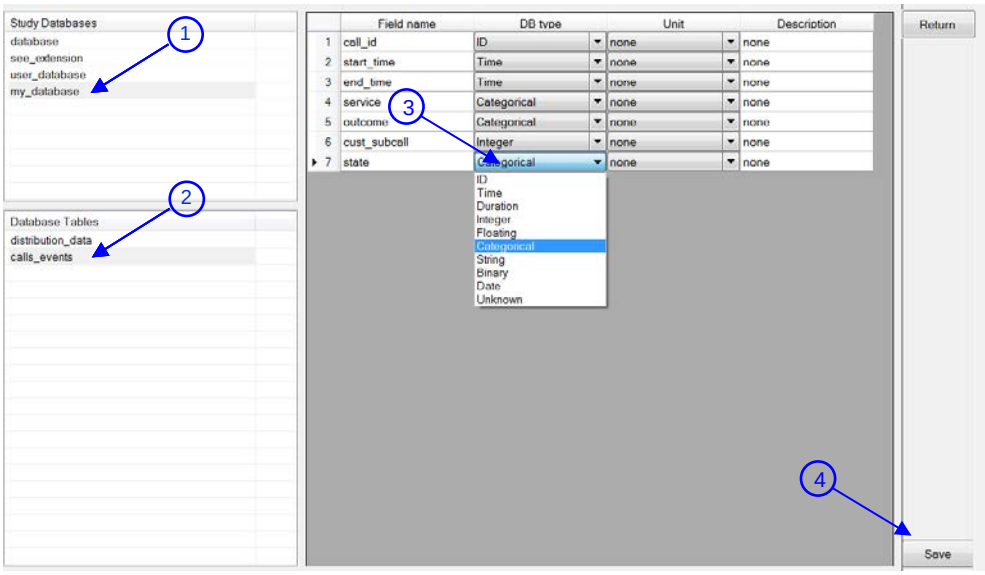
Click on operator *left bracket* (**(** **)** in Operators/functions list box. Select *duration* field in Field name list box and click on operator *right bracket* (**)** **)**

Appendix 6 How to change type of field

Click *Main->Data Management->Database Management-> DB Tables Editing*



Select study (if needed).



(1) Select database
(2) Select database table
(3) Click on DB type and chance DB type
(4) Click Save Button

Appendix 7 Calculations (DRAFT)

State Animation:
Calculate instants (moment of time) counts in time interval from start_time to end_time for **current** event.

Hybrid Animation:
Compute instants (moment of time) counts in time interval from start_time of **current event** to start_time for **next event**.

Instants calculation

Compute instants (moment of time) counts in time interval from start_time to end_time.
Define:

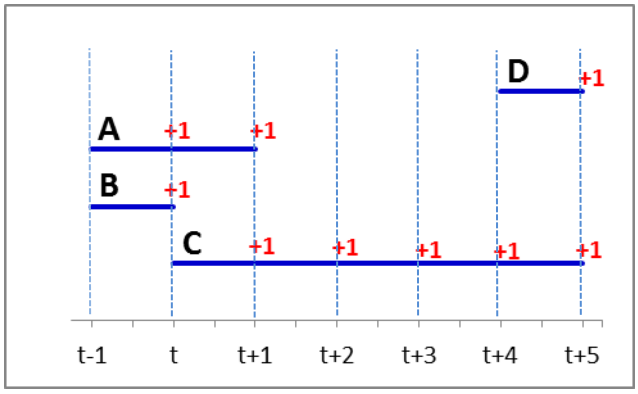
- **Discrete scale of time**: 24 hours of day divided into 86400 one second length time intervals.
- **m_j** – moment of time j, or second$_j$; j=1, 2, …, nt;
  nt – length of time interval or number of moments of time;

  nt = end_time – start_time + 1
- **x_i** – event i (for example call waited in queue or patient gets treatment in ED);
  i=1, 2, … , n;
  n – total number of events
- **x_{iopen}** – "open" event i at moment of time, so that event start > start_time and event end ≤ end_time
- **$N_{at\ moment\ i}$** – number of "open" events at moment of time i

  $N_{at\ moment\ i} = \sum_i^n x_{iopen}$

Events A, B, C, D were observed at time interval from t–1 to t+5. Event start time and event end time known.

event	event start	event end
A	t–1	t+1
B	t–1	t
C	t	t+5
D	t+4	t+5



+1 – *add "open" event at moment of time*

Questions: How many "open" events were at every moment of time in defined interval?

Let see the sample:
n = 4
start_time = t–1
end_time = t+5
nt = t+5– (t–1)+1 = 7 moments of time

j	m_j	$N_{at\ moment\ i}$	x_{iopen}
1	t–1	0	
2	t	2	A, B
3	t+1	2	A, C
4	t+2	1	C
5	t+3	1	C
6	t+4	1	C
7	t+5	2	C, D