

SOUTENANCE DE STAGE

CONNEXION À UN ROBOT ET VISUALISATION DE SA LOGIQUE INTERNE

JUSTINE FRICOU

3E ANNÉE

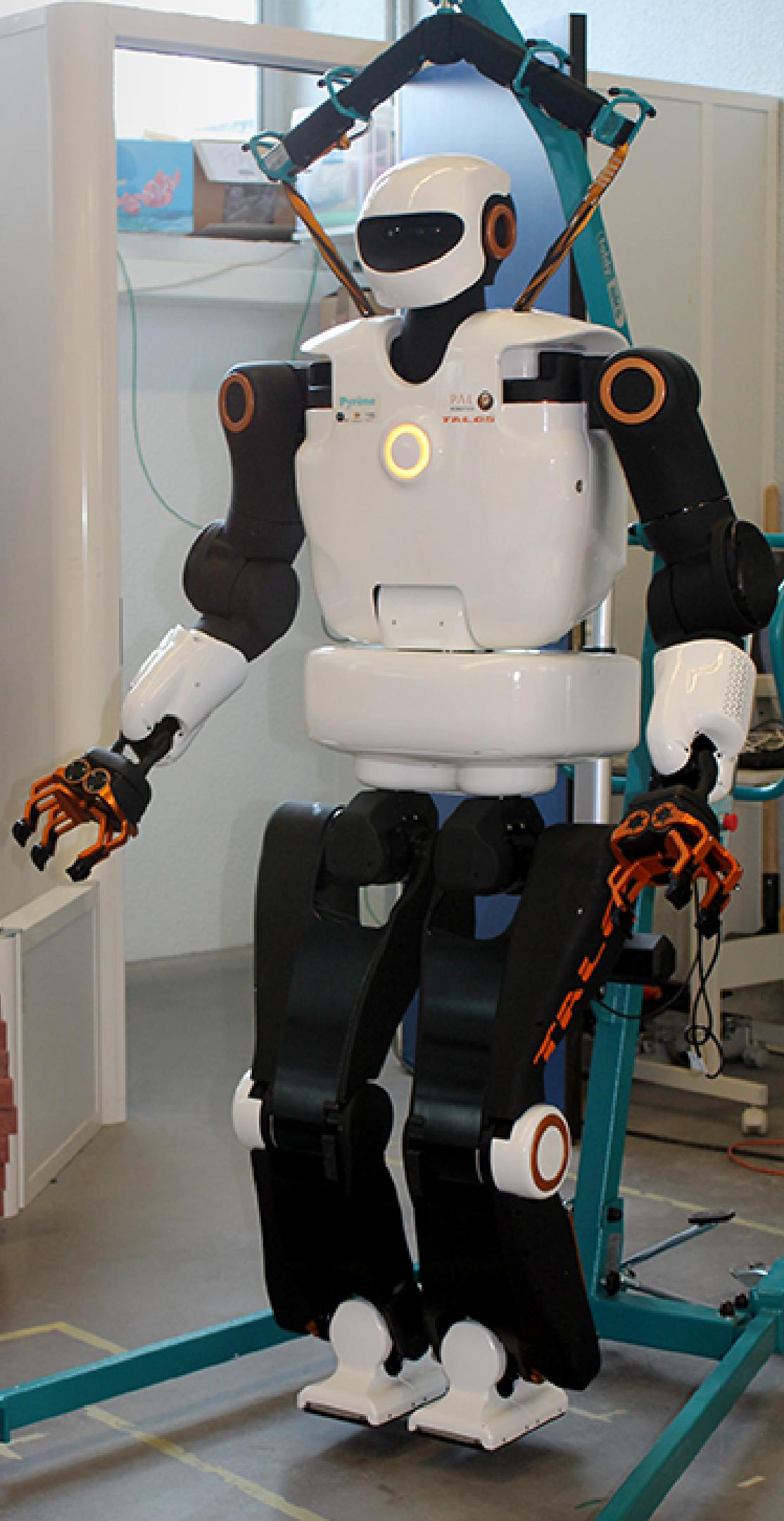
PGE 2024

LAAS-CNRS



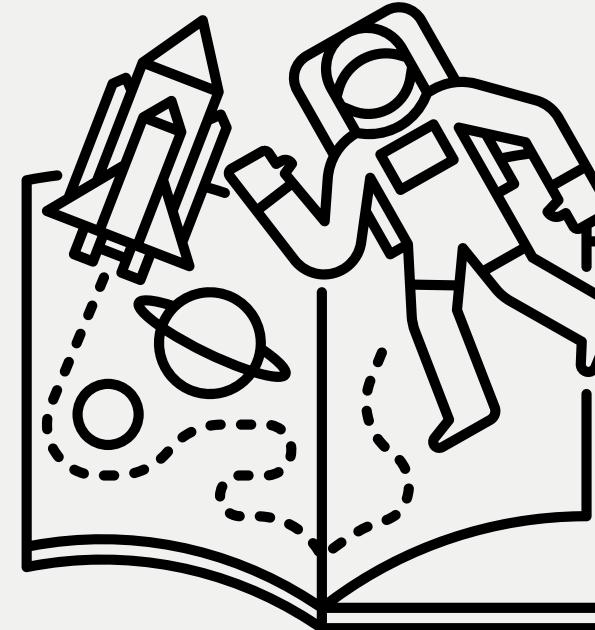
TUTEURS :
FLORENT LAMIRAUD &
MAXIMILIEN NAVEAU

EPITECH TOULOUSE



MISE EN CONTEXTE : POURQUOI CE STAGE ?

Ma vocation



Recherche
scientifique

Informatique

-> Ingénieure de recherche

Le LAAS-CNRS



Institution publique

Biologie, mathématiques, sciences sociales, etc

"Faire progresser la connaissance et être utile à la société"

Gestion de l'énergie, robotique, optique, etc

Nombreux partenariats avec le monde de l'industrie

Applications à la santé, l'espace, l'environnement, les transports...



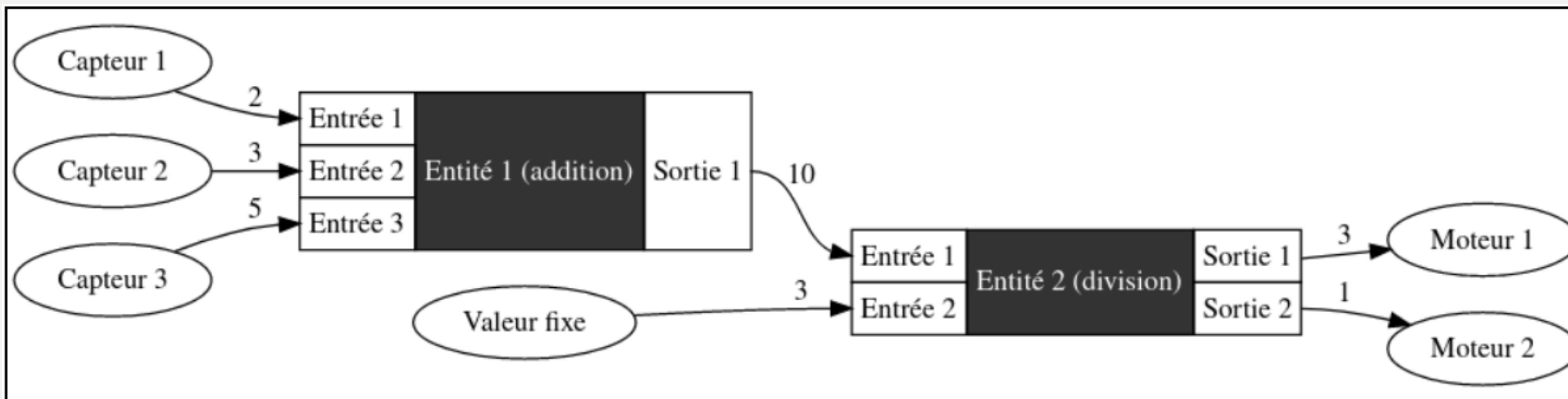
L'équipe Gepetto



- . Mouvement humain
- . Muscles artificiels
- . Planification et commande de mouvements robotiques

MA CONTRIBUTION

La 'Stack of Tasks'

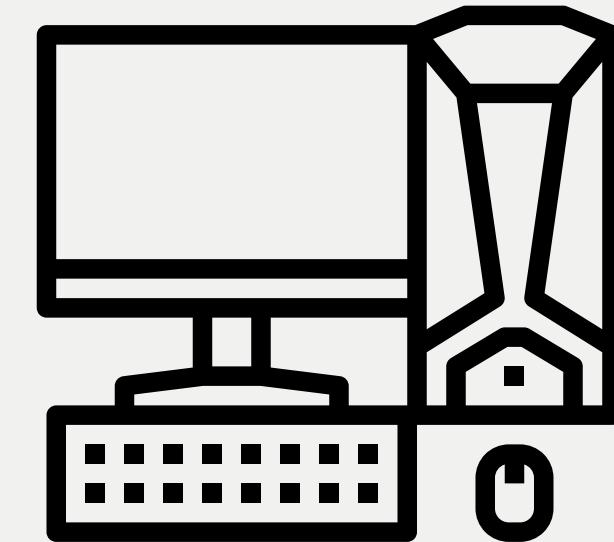
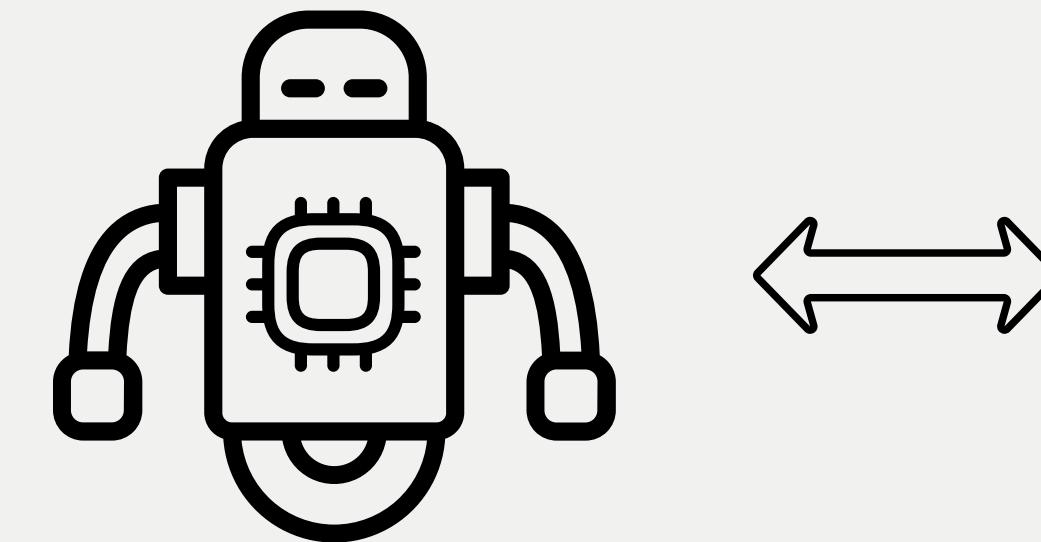


Exemple simplifié de graphe de contrôle

Ma contribution

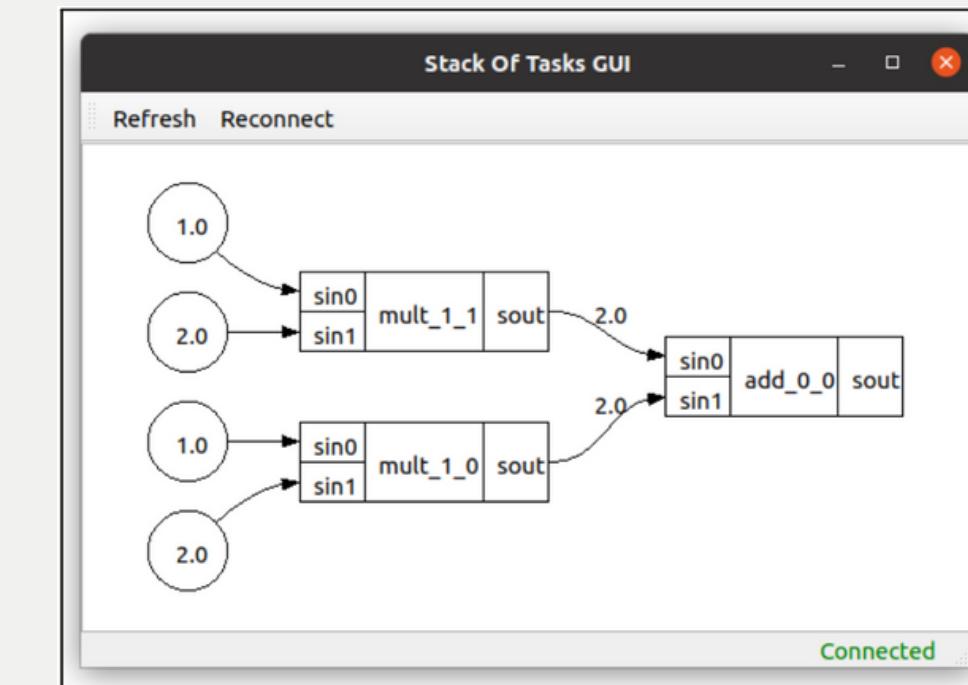
sot-ipython-connection

Connexion au robot
-> flexibilité



sot-gui

Visualisation de la SoT
-> affichage détaillé
-> lisibilité

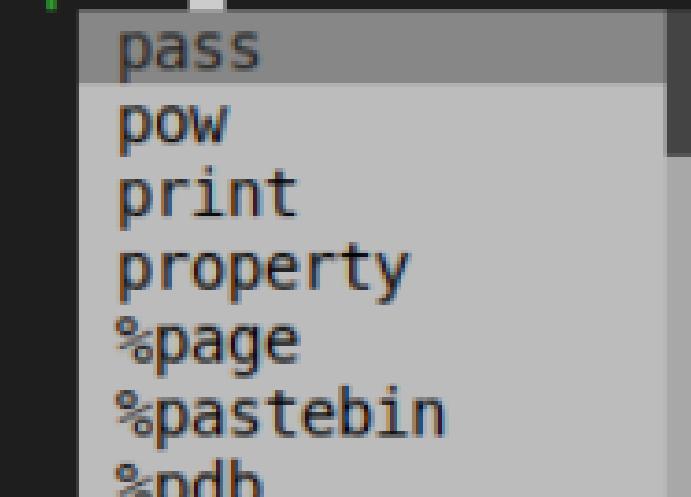


sot-ipython-connection : ipython

```
NOTE: When using the `ipython kernel` entry point, Ctrl-C will not work.  
To exit, you will have to explicitly quit this process, by either sending  
"quit" from a client, or using Ctrl-\ in UNIX-like environments.  
To read more about this, see https://github.com/ipython/ipython/issues/2049  
To connect another client to this kernel, use:  
--existing kernel-37652.json
```

Kernel

```
In [1]: 1 + 1  
Out[1]: 2  
  
In [2]: pass
```



Client

sot-ipython-connection : fonctionnalités

```
connection_config = dict([
    shell_port = 42767,
    iopub_port = 46117,
    stdin_port = 38481,
    control_port = 9872,
    hb_port = 52277,
    ip = "127.0.0.1",
    transport = "tcp",
    signature_scheme = "hmac-sha256",
])
```

Configuration de la connexion

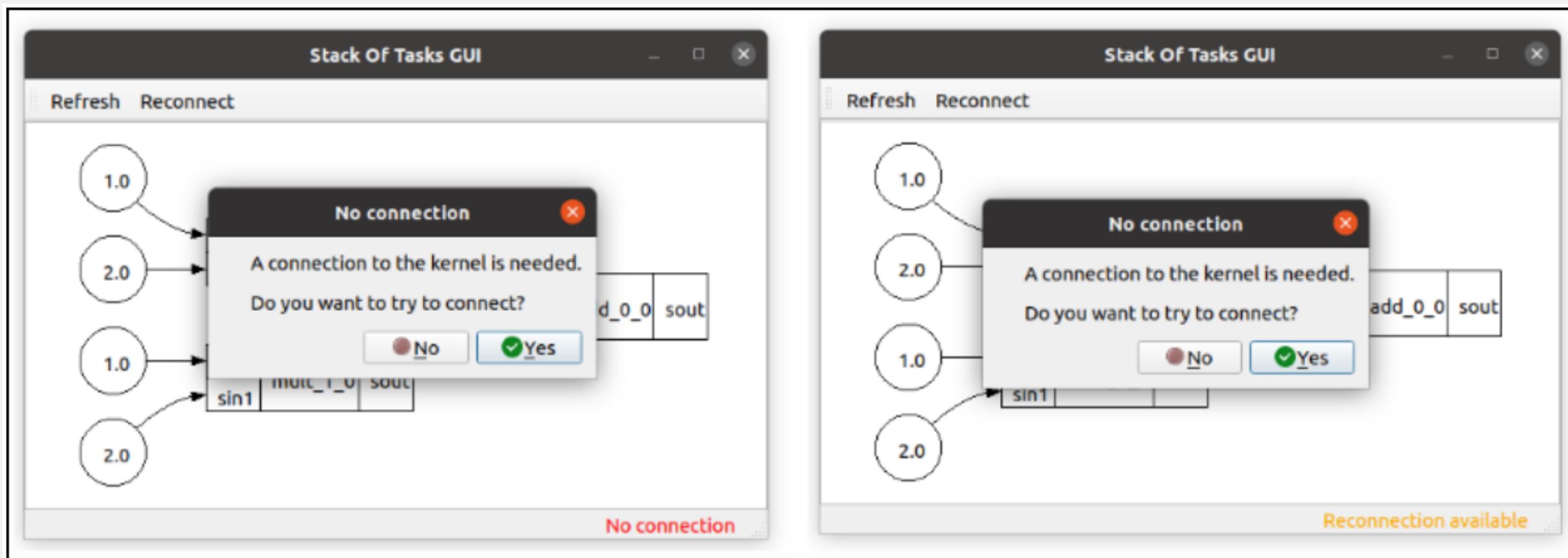
```
kernel_namespace = dict([
    initial_namespace_1 = 46,
    initial_namespace_2 = 54
])
```

Namespace initial

```
>>> client.print_history()
Session id: 77c6a5fd-dd805ac7db5479a83d8faed1
Command id: 77c6a5fd-dd805ac7db5479a83d8faed1_48532_0
Command: 1+1
Result: 2 (<class 'int'>)
```

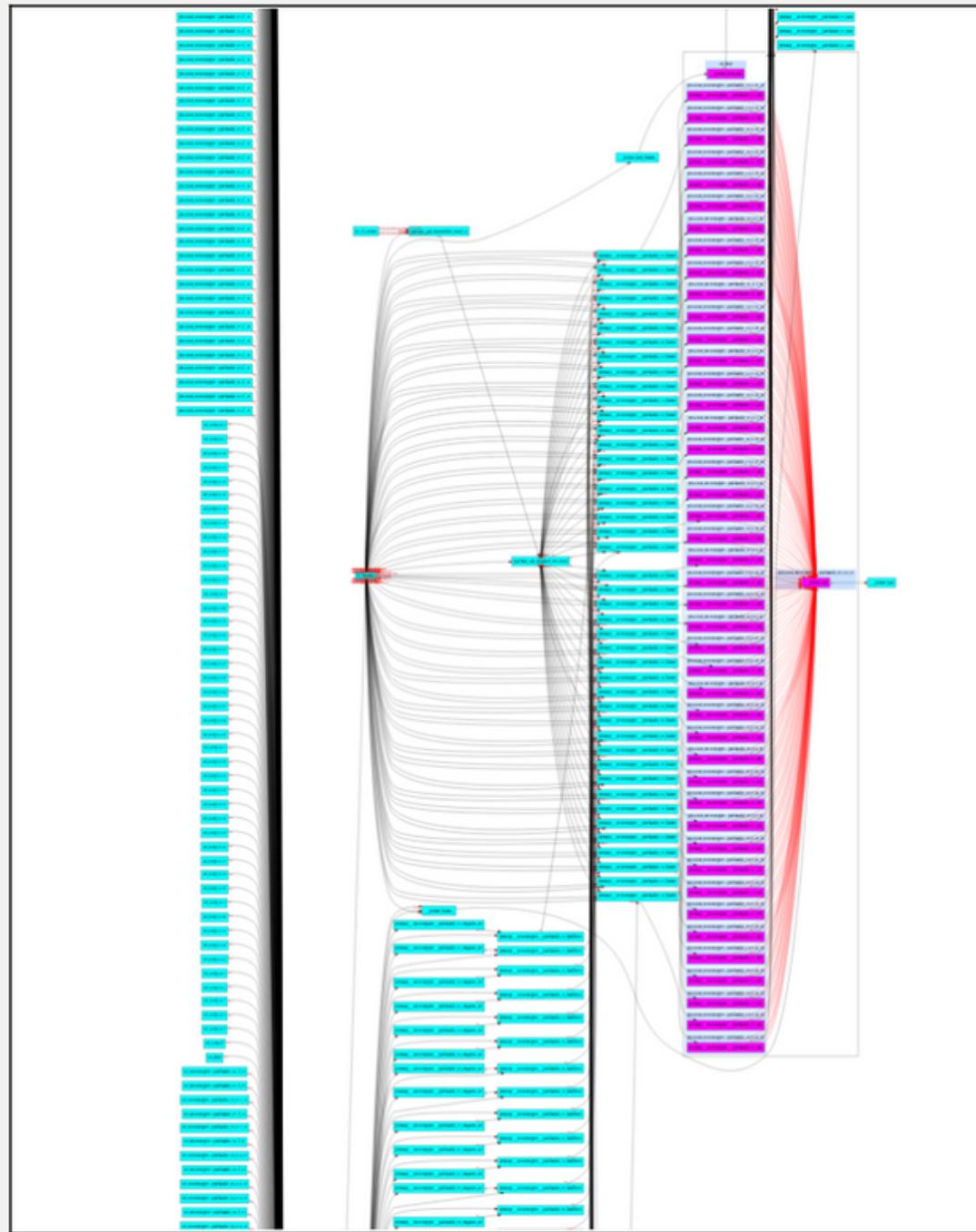
Historique

sot-ipython-connection : exemple d'application

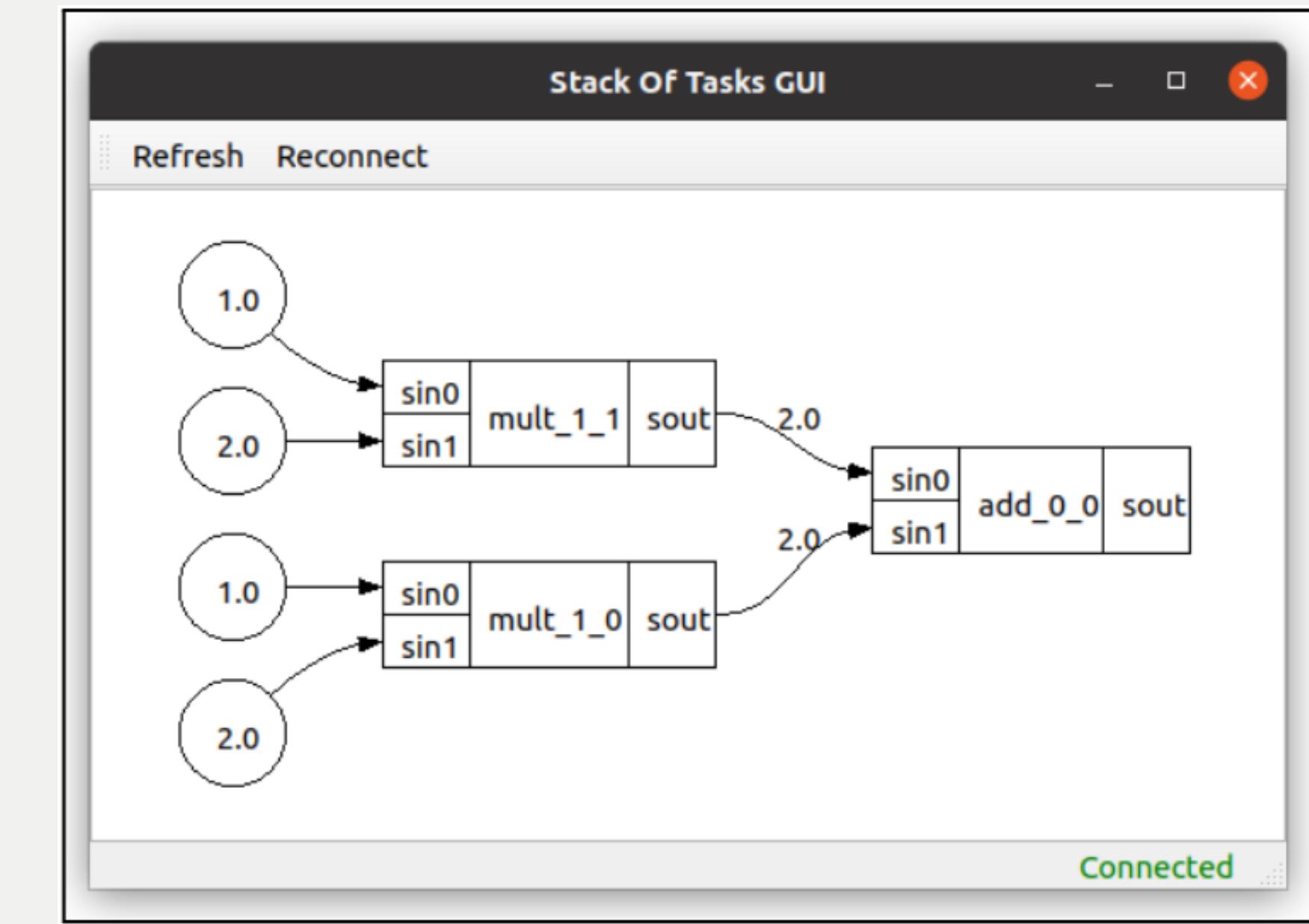


Alerte sur le statut de connexion

sot-gui

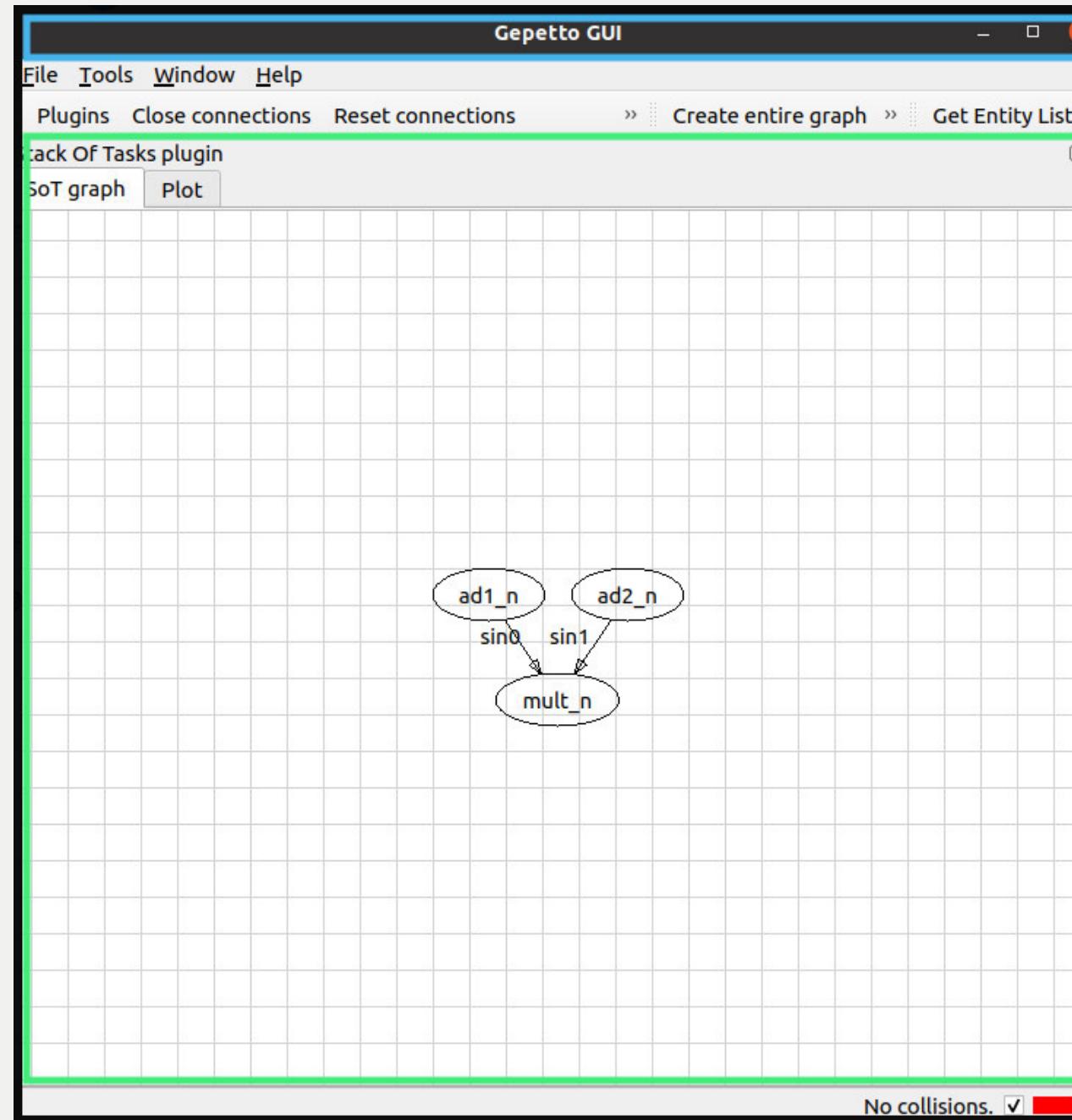


Avant : pdf, png... (statique)

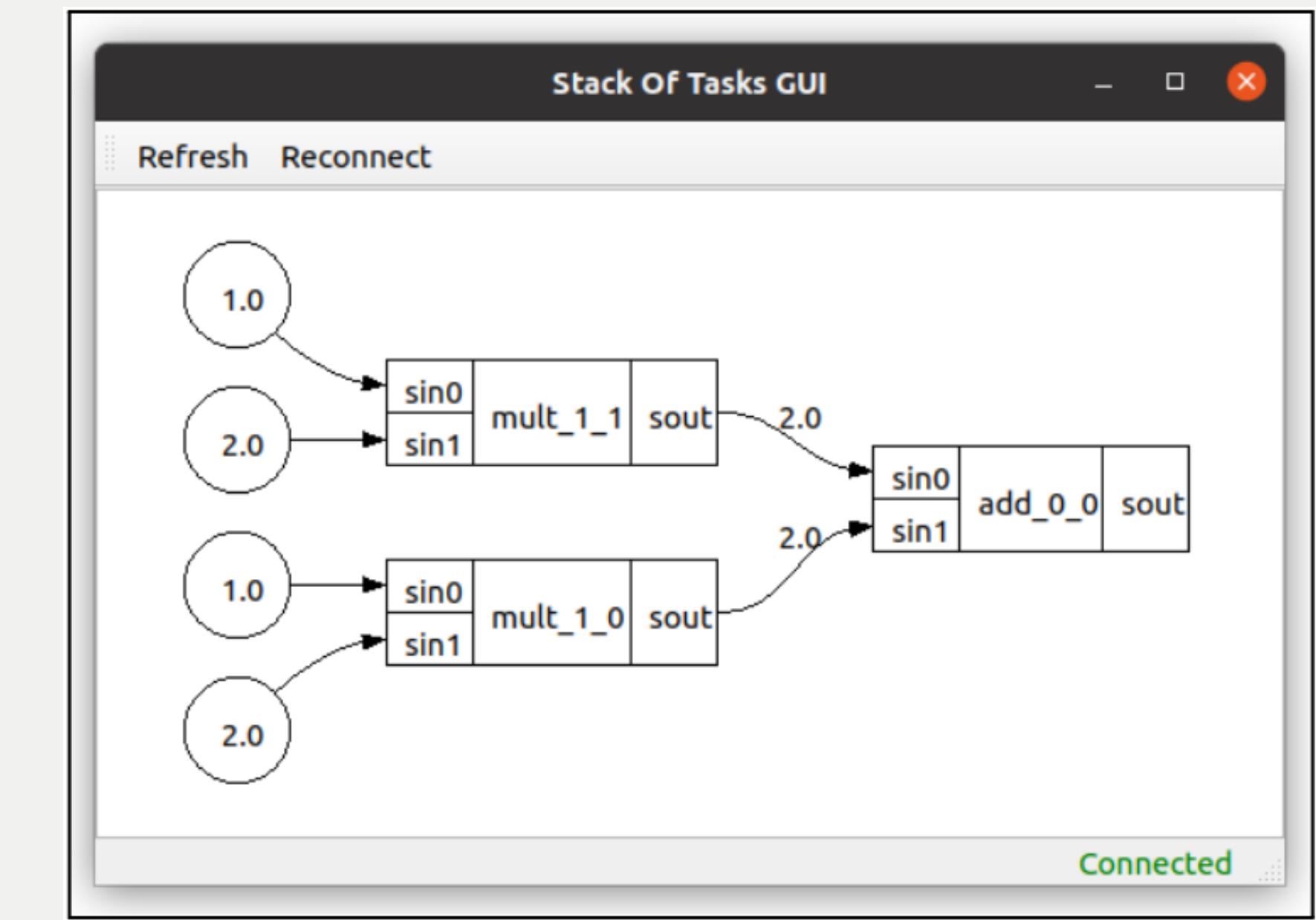


Après : sot-gui (dynamique)

sot-gui



Avant : sot-gepetto-viewer (obsolète)



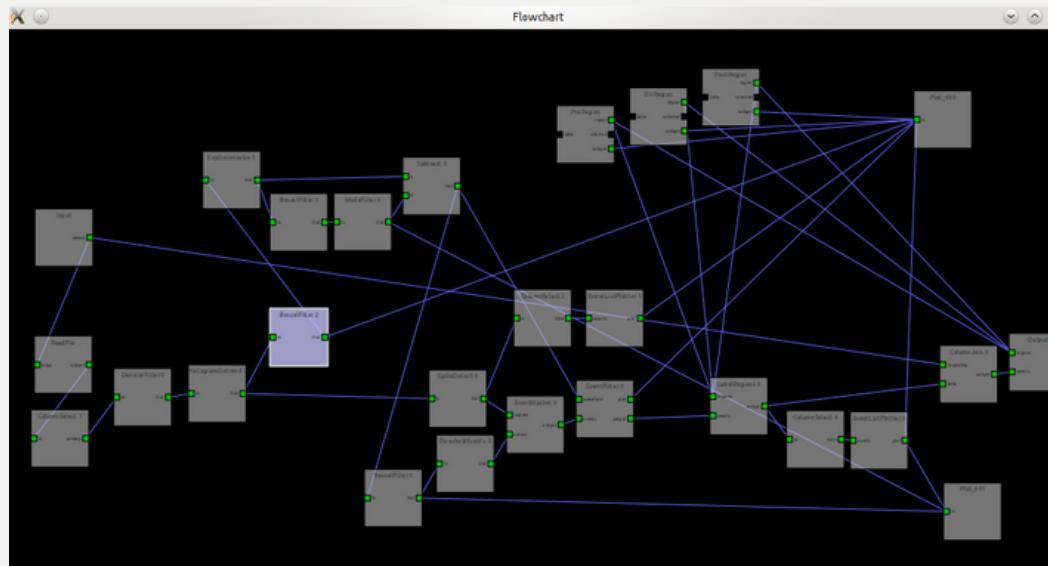
Après : sot-gui (dynamique)

sot-gui : recherche d'outils

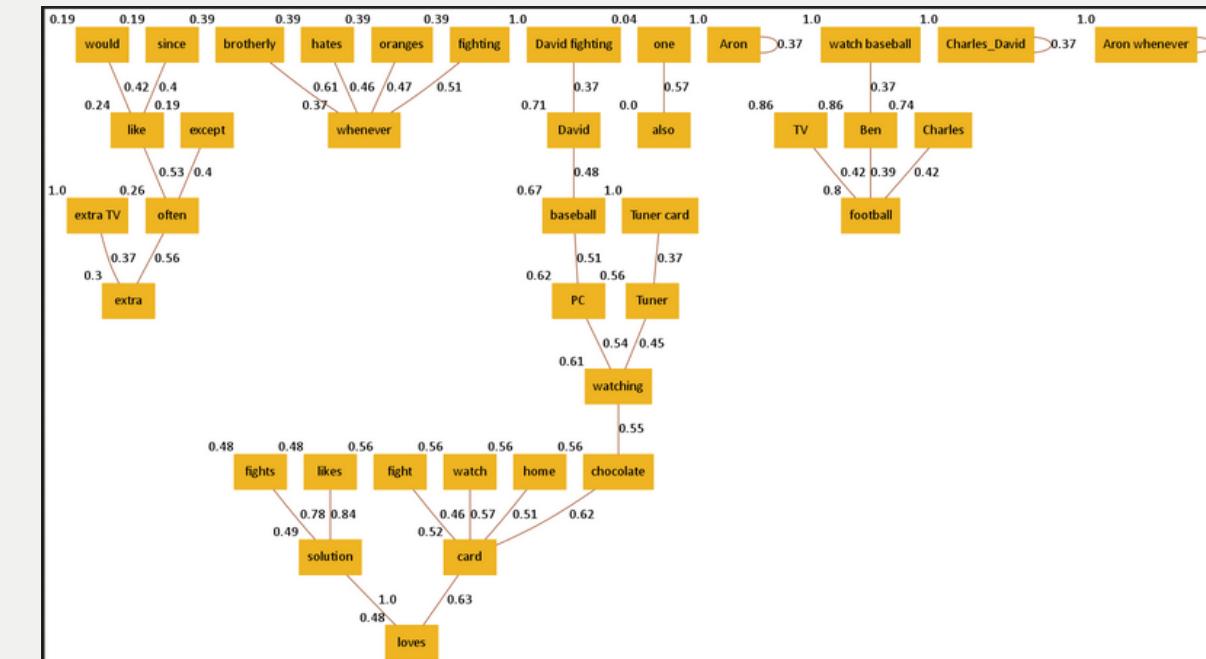
Les critères

- . Affichage de graphe
- . Le moins de dépendances possibles
- . Dépendances open source
- . Intégration à Gepetto GUI et rqt : utiliser python et Qt

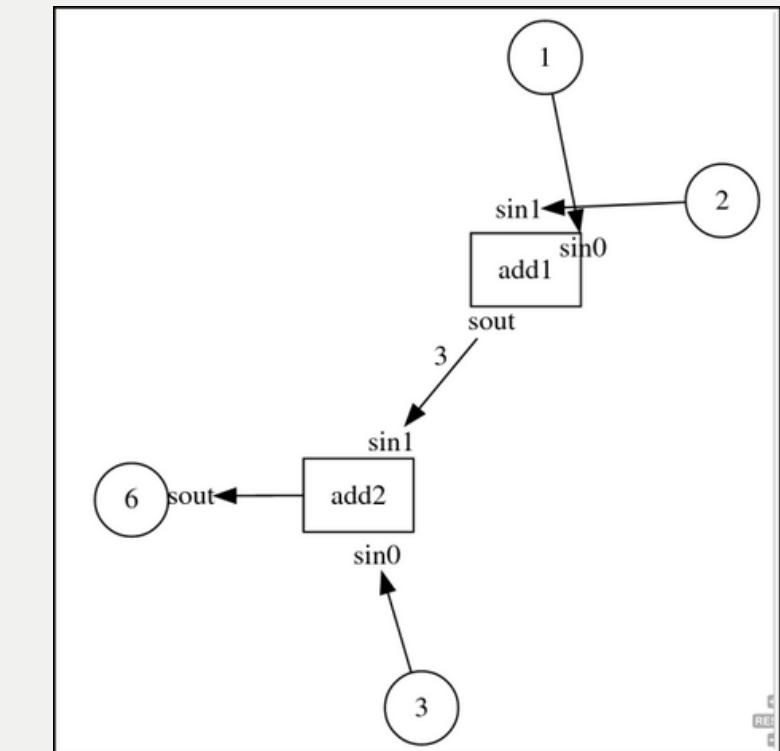
sot-gui : recherche d'outils



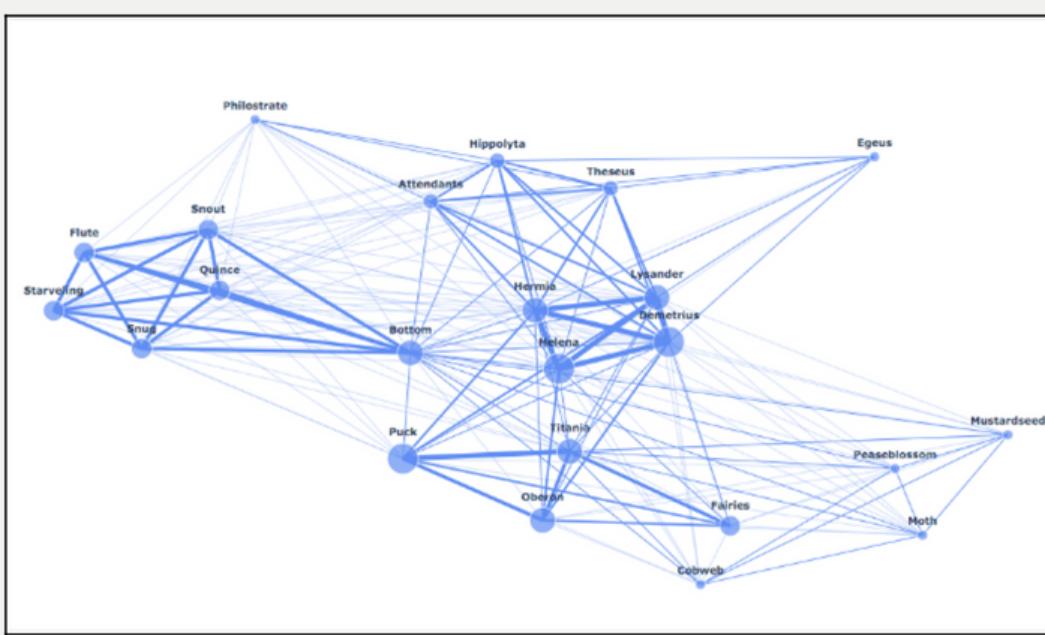
PyQtGraph



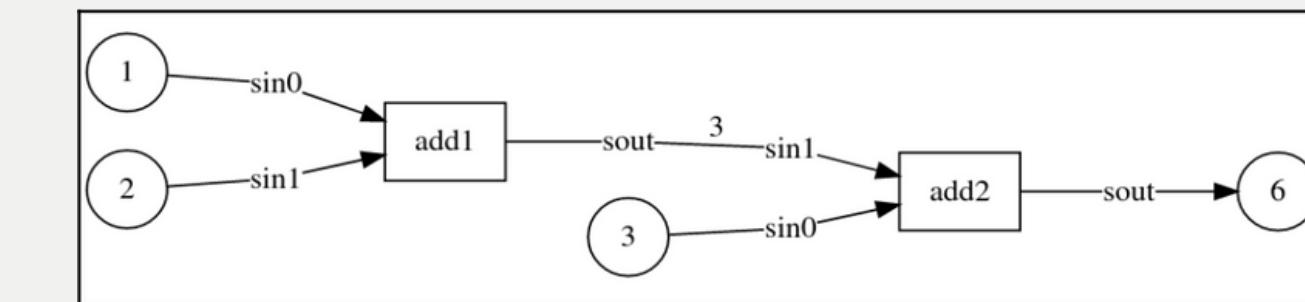
PyGraphViz



Graphviz (neato)

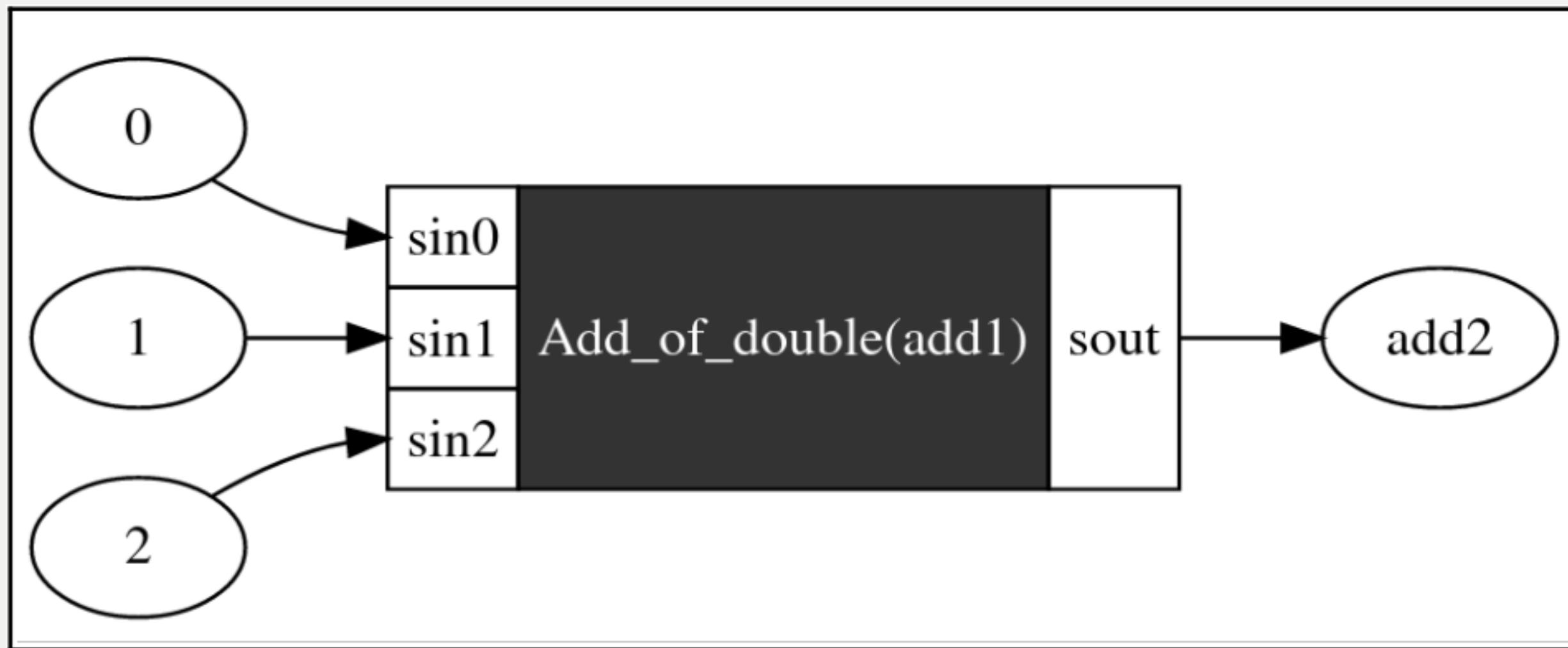


Networkx



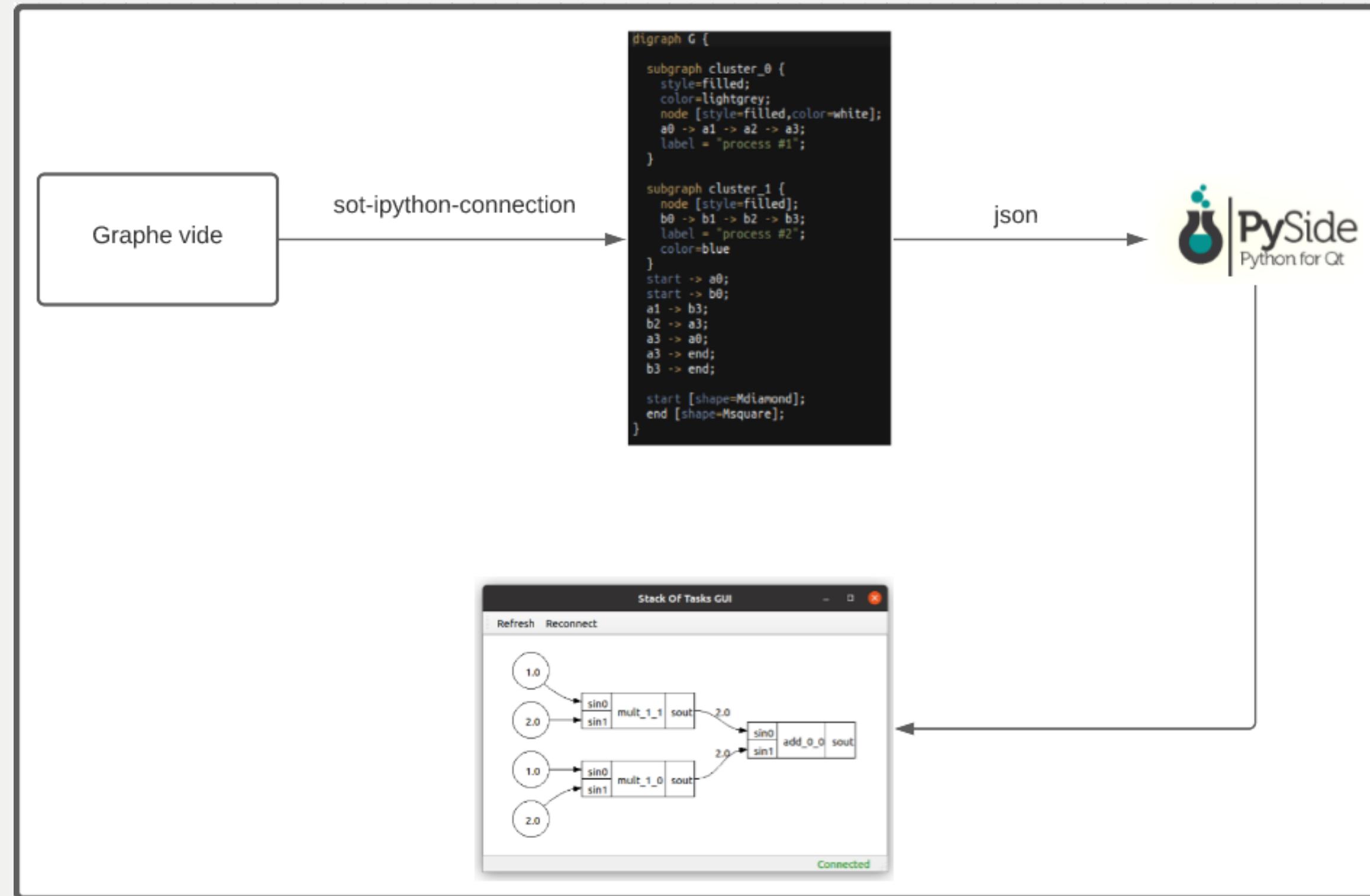
Graphviz (dot)

sot-gui : recherche d'outils



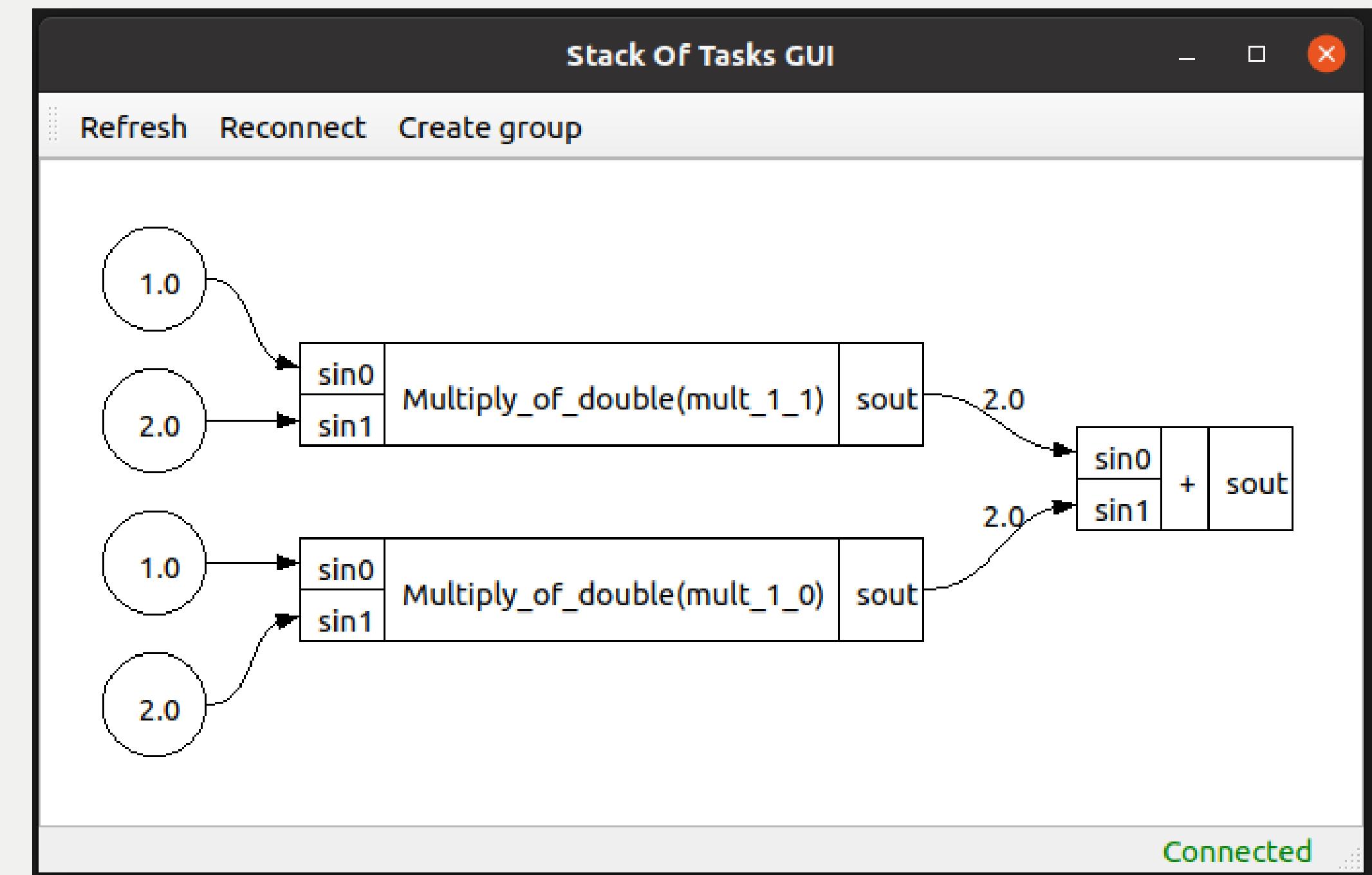
Graphviz (dot + tables html)

sot-gui : structure globale

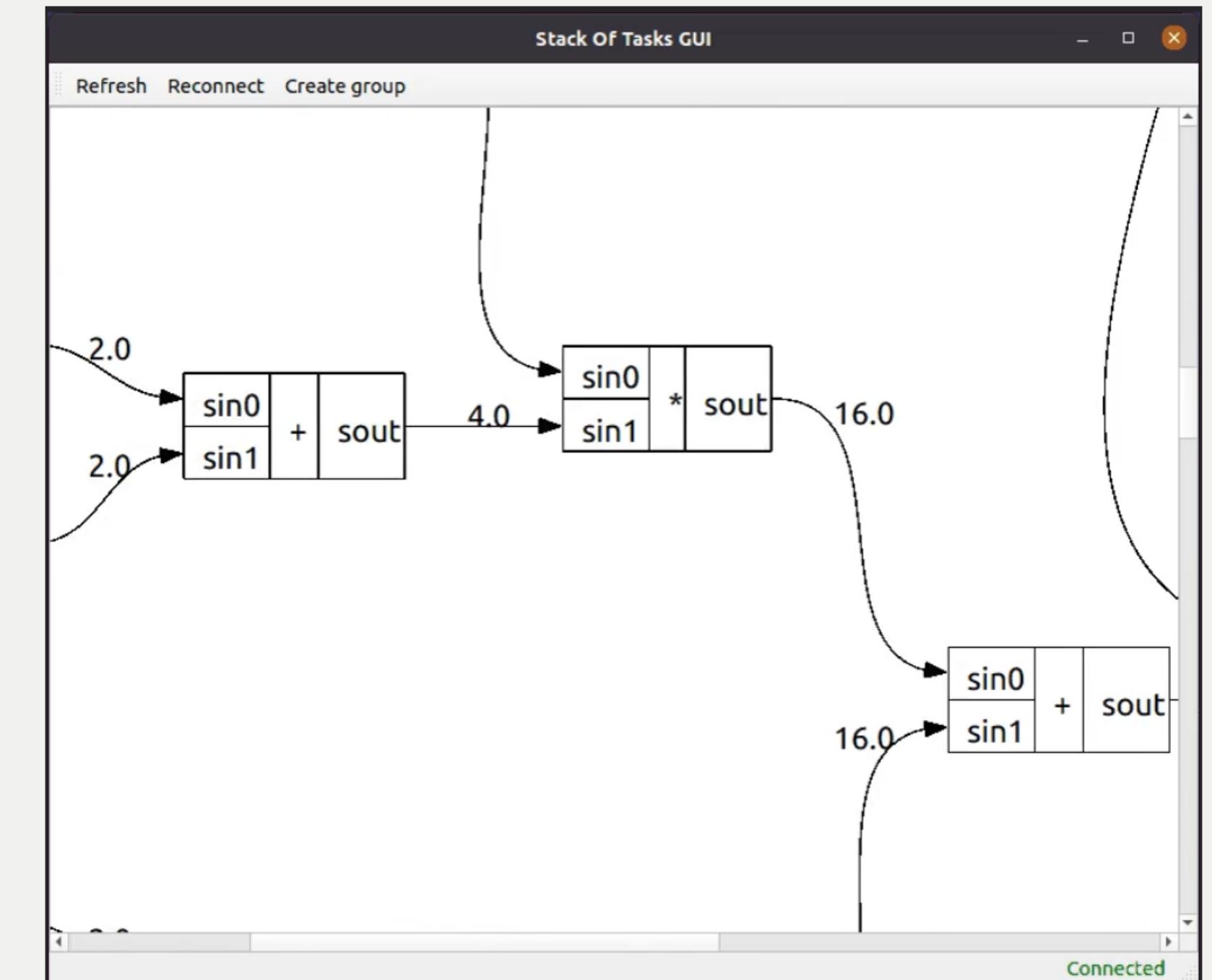
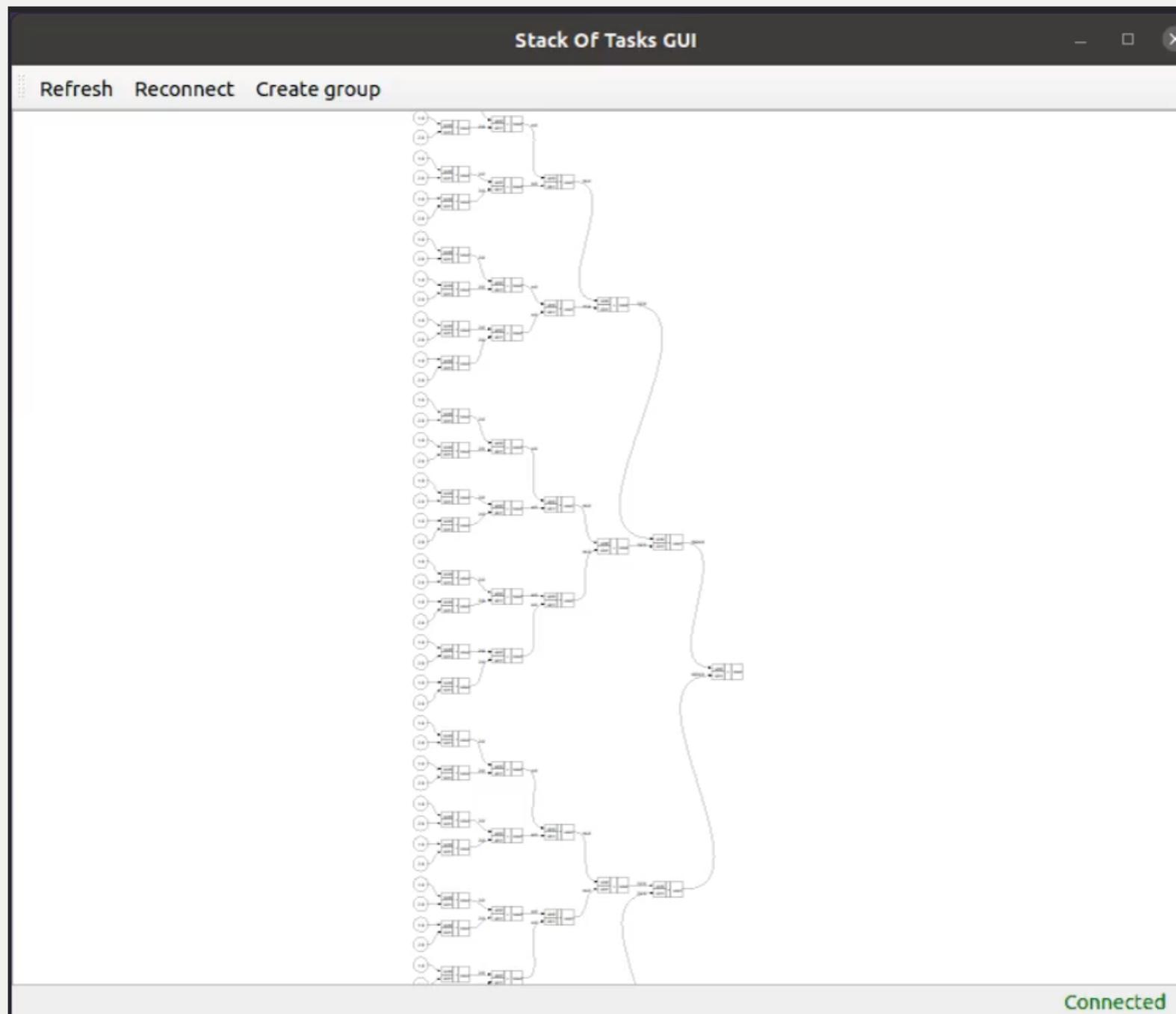


sot-gui : alias pour les entités

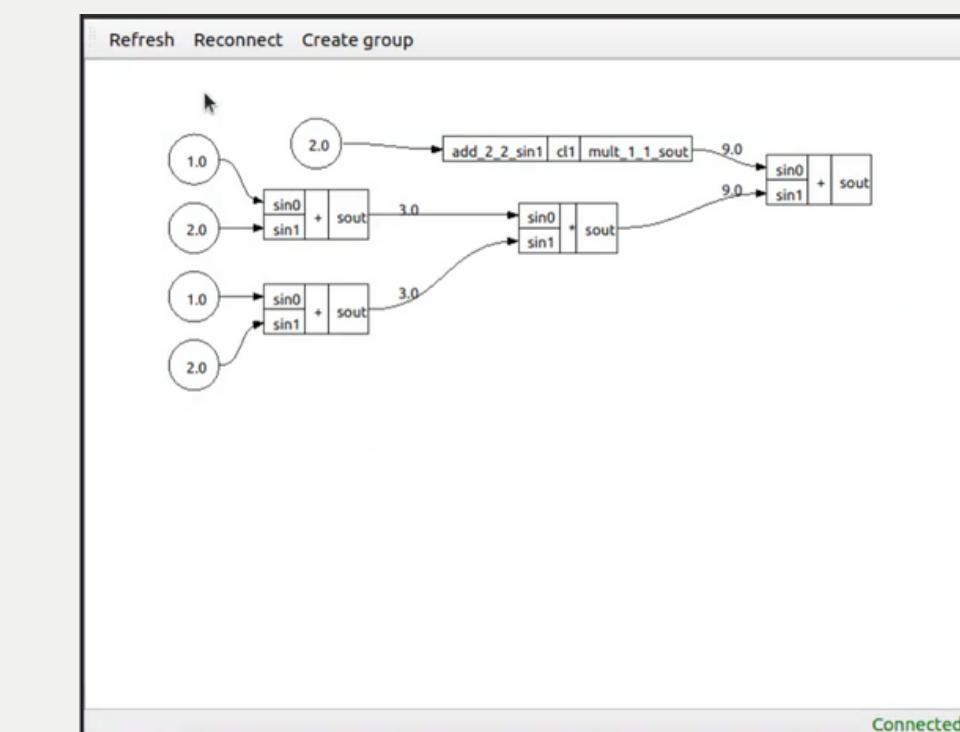
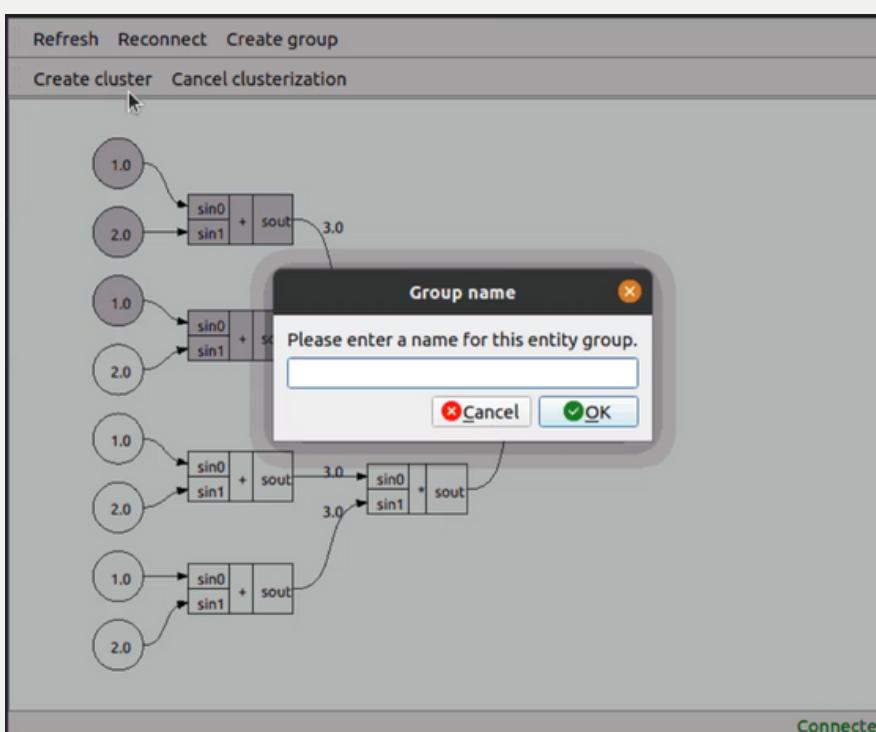
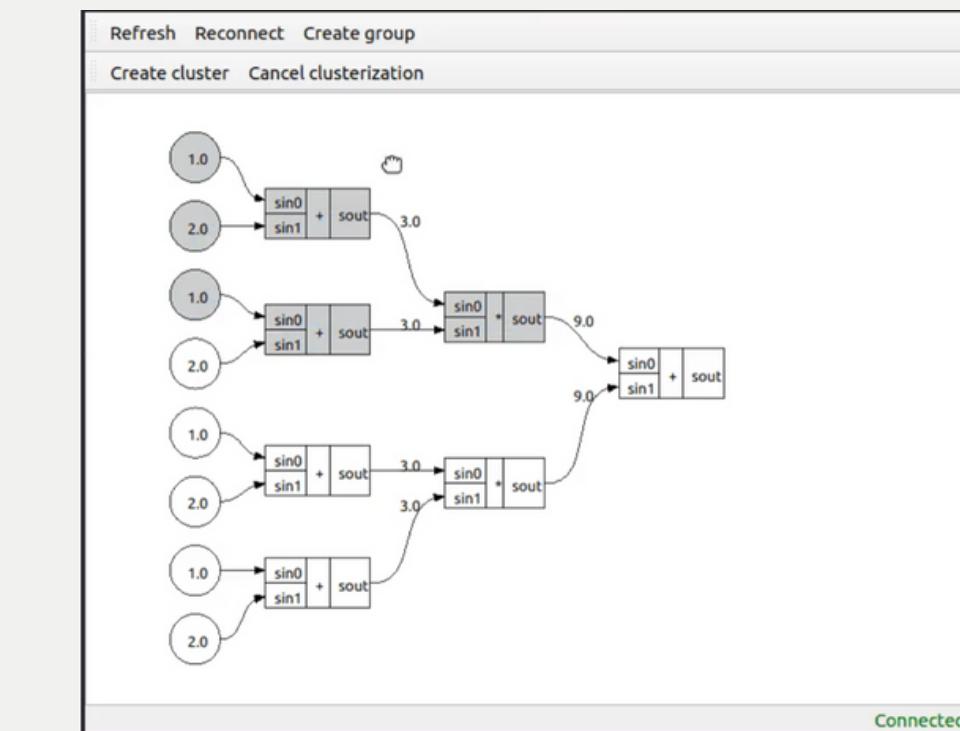
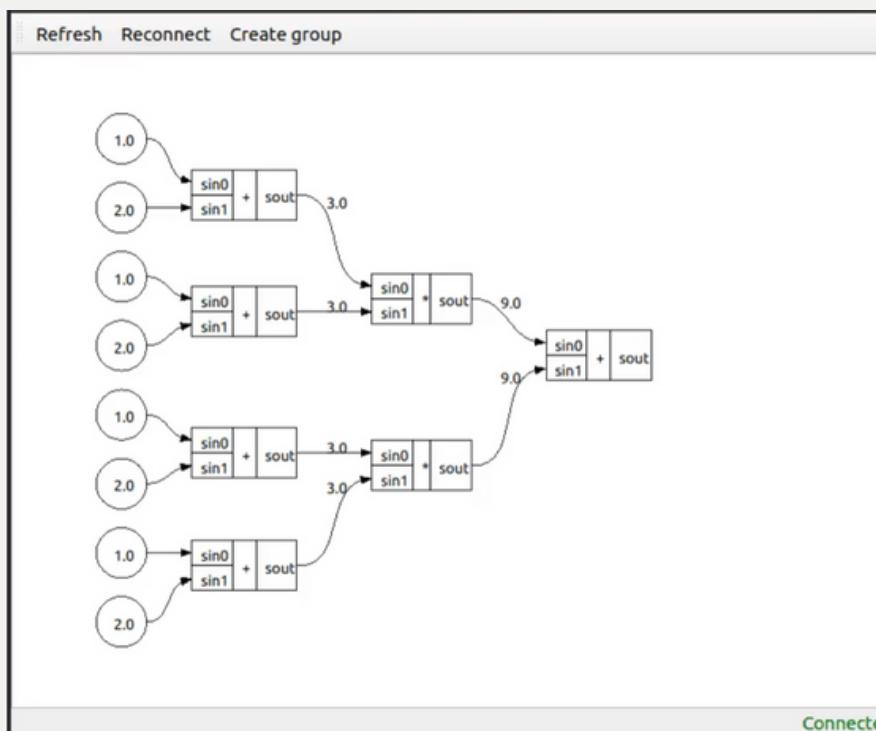
```
entities_labels = {
    'Add_of_double': '+',
}
```



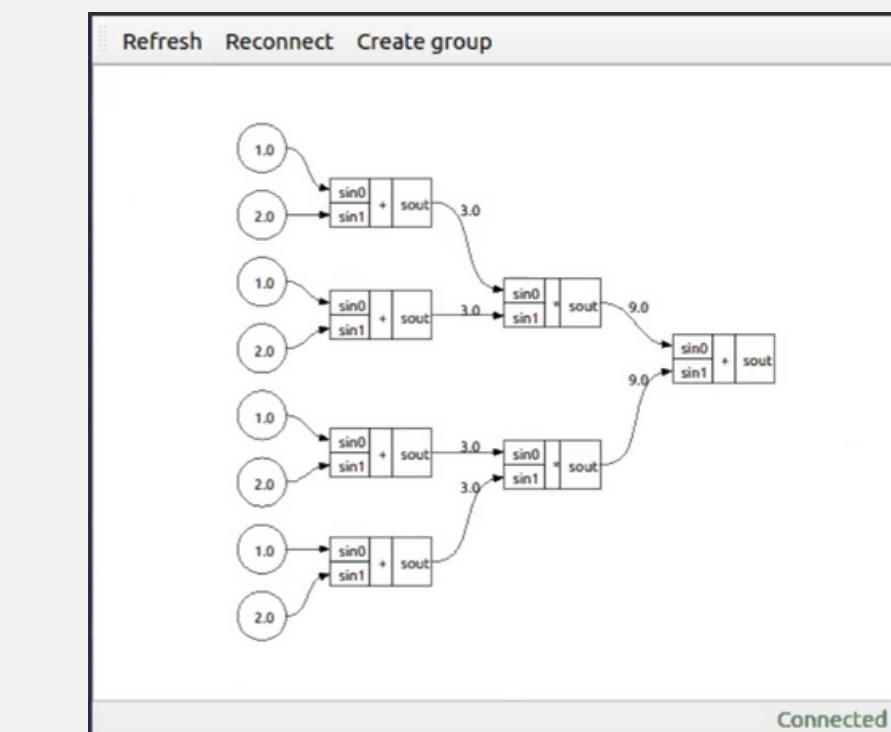
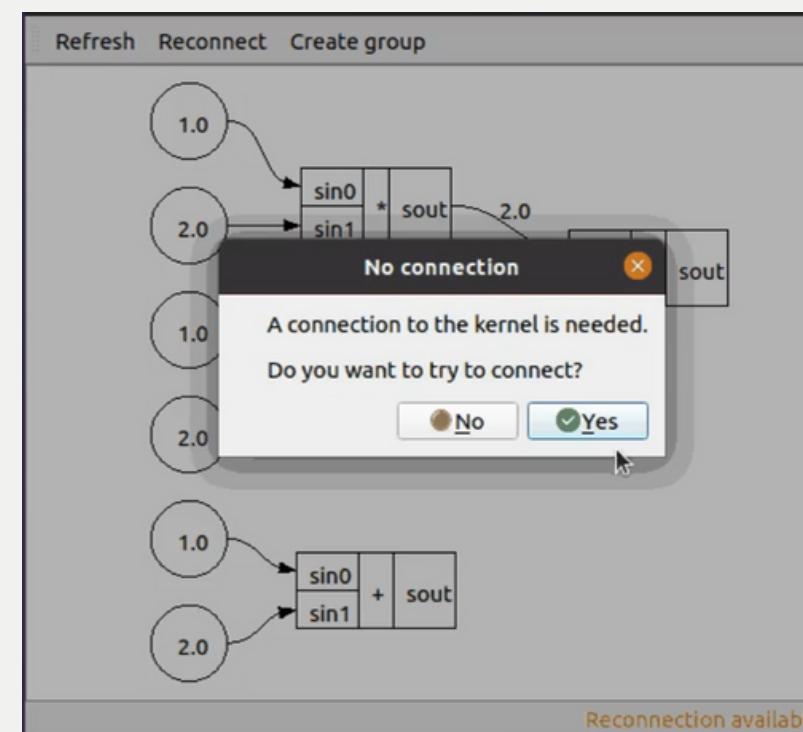
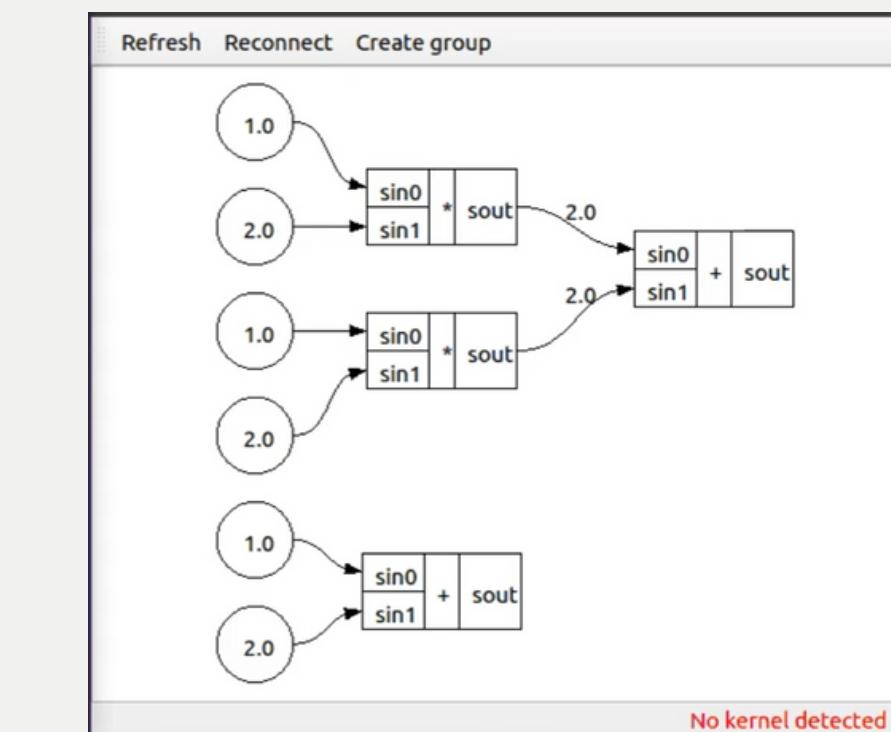
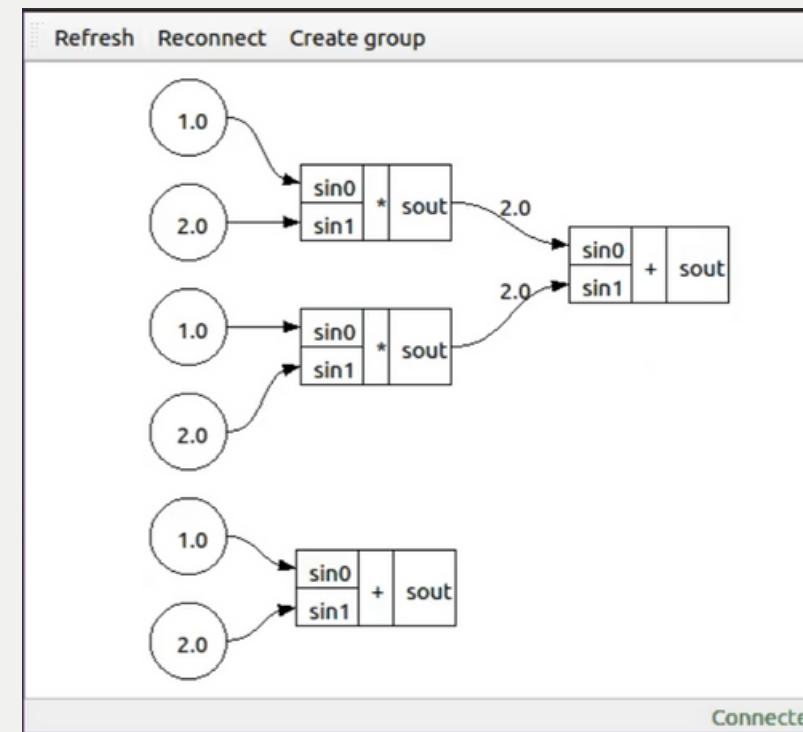
sot-gui : zoom



sot-gui : groupement d'entités



sot-gui : actualisation



Documentation et tests

```
class DotDataGenerator(builtins.object)
    DotDataGenerator(graph_name: str = 'G')

This class allows to generate dot code through a simple API.

Methods defined here:

__init__(self, graph_name: str = 'G')
    Initialize self. See help(type(self)) for accurate signature.

add_edge(self, tail: Tuple[str, str], head: Tuple[str, str], attributes: Dict[str, Any] = None) -> None
    Adds an edge to the graph, with optional attributes.

Args:
    tail: tuple containing the tail data: (node name, port name).
        The port name can be None.
    head: tuple containing the head data: (node name, port name).
        The port name can be None.
    attributes: A dictionary containing the attributes of the edge.
        The keys and values are in the same form as in dot code (e.g
            'attributes['color'] = 'red'' corresponds to 'color=red' in dot.
```

get_qt_item_for_node(self, node_name: str, no_input: bool = False) -> PySide2.QtWidgets.QGraphicsItem

Returns a qt item corresponding to a node.

The qt item will be the node's body (i.e its shape OR the middle column of the html table) as the parent QGraphicsItem, which will contain the node label as a child item.
All those items' positions will be set before return.
If the node's style was set to invisible, this method will return None.

Args:

- node_name: name of the node, as given in the dot code used to generate the json output.
- no_input: True if the node has no input port displayed.

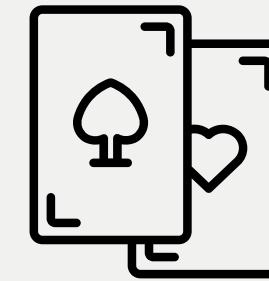
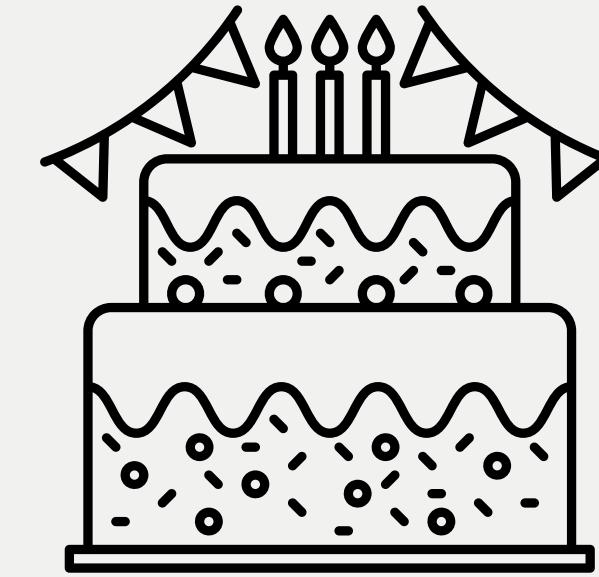
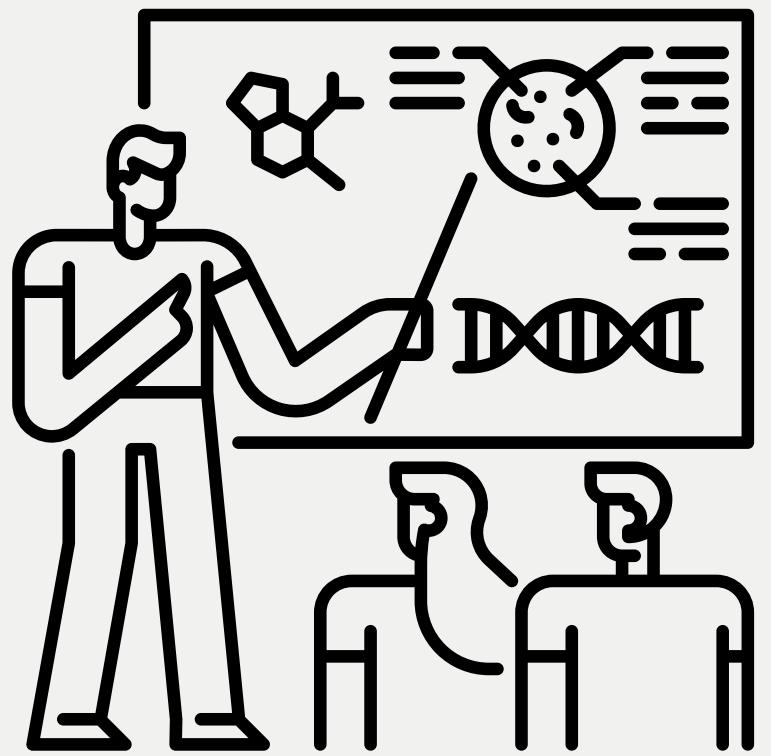
Documentation de sot-gui

Name	Stmts	Miss	Cover
/home/jfricou/projects/workspace/install/sot_ipython_connection/lib/python3.8/site-packages/sot_ipython_connection/__init__.py	0	0	100%
/home/jfricou/projects/workspace/install/sot_ipython_connection/lib/python3.8/site-packages/sot_ipython_connection/app/__init__.py	0	0	100%
/home/jfricou/projects/workspace/install/sot_ipython_connection/lib/python3.8/site-packages/sot_ipython_connection/app/sot_script_executer.py	19	1	95%
/home/jfricou/projects/workspace/install/sot_ipython_connection/lib/python3.8/site-packages/sot_ipython_connection/connection_config.py	1	0	100%
/home/jfricou/projects/workspace/install/sot_ipython_connection/lib/python3.8/site-packages/sot_ipython_connection/kernel_namespace_config.py	1	0	100%
/home/jfricou/projects/workspace/install/sot_ipython_connection/lib/python3.8/site-packages/sot_ipython_connection/sot_client.py	172	52	70%
/home/jfricou/projects/workspace/install/sot_ipython_connection/lib/python3.8/site-packages/sot_ipython_connection/sot_kernel.py	46	9	80%
src_python/sot_ipython_connection/__init__.py	0	0	100%
src_python/sot_ipython_connection/sot_client.py	172	56	67%
TOTAL	411	118	71%

Pourcentage de couverture de sot-ipython-connection avec Pytest

MON BILAN

L'équipe



Pour la suite

4e année

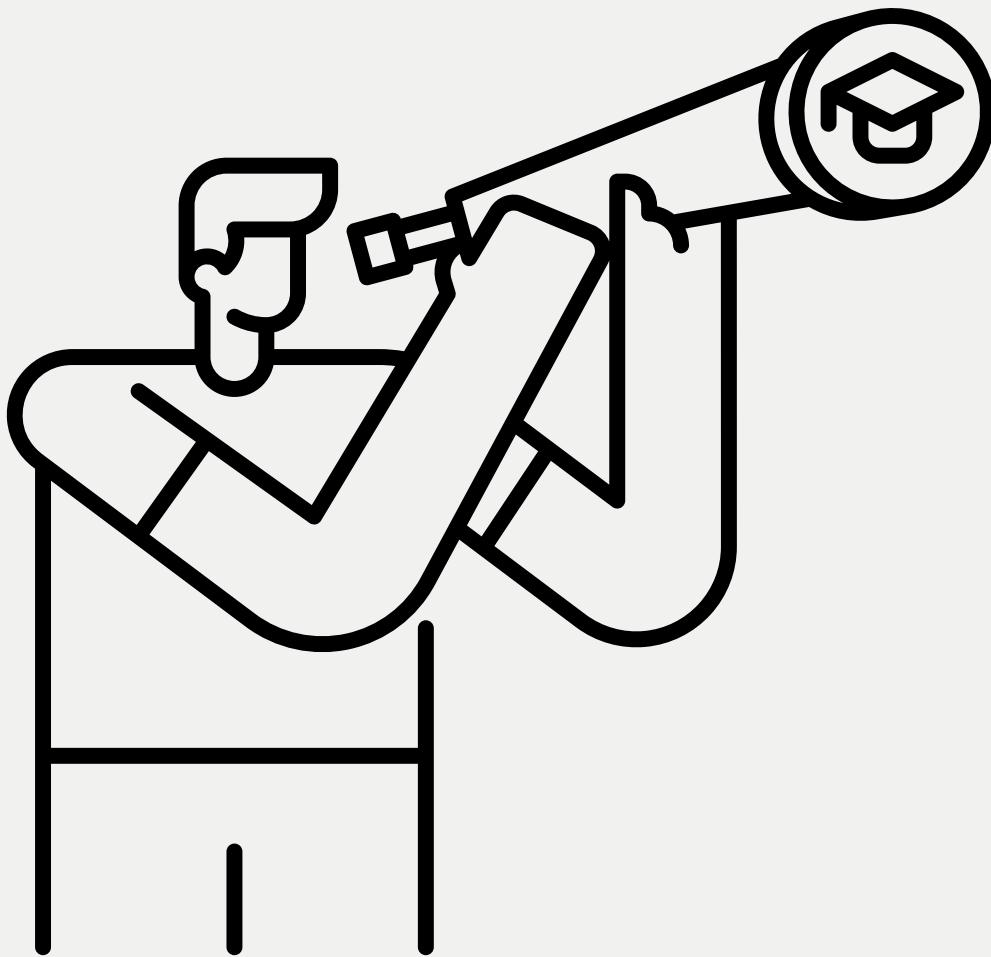
Modules nouveaux et variés

-> polyvalence

Stage de fin d'études

Nouveau domaine de recherche ?

-> explorer les possibilités



Avez-vous des questions ?

Crédits

Images : laas.fr, pyqtgraph.org, Rebecca Weng sur towardsdatascience.com

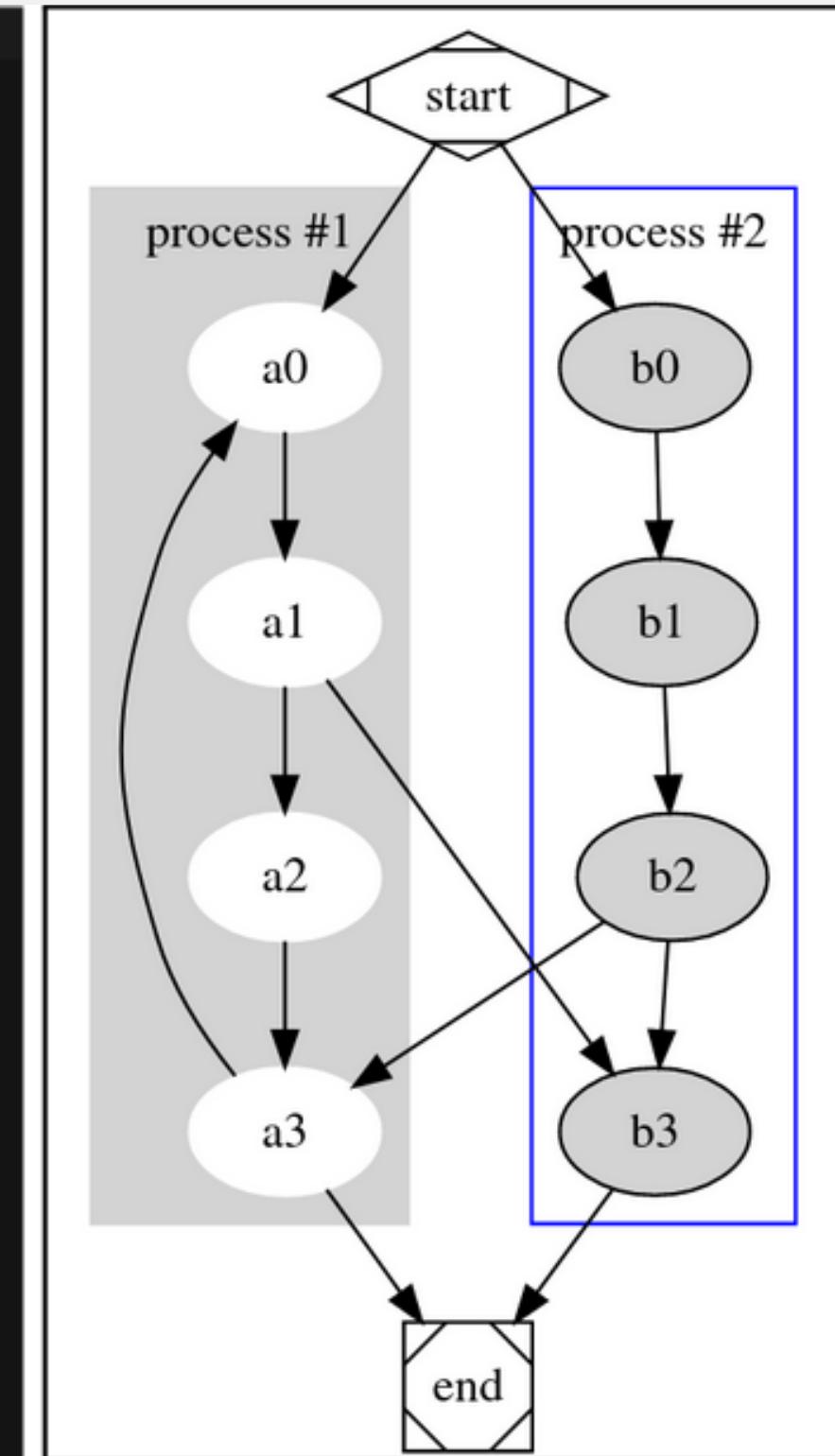
Icônes : Chattapat et iconnut sur thenounproject.com


```

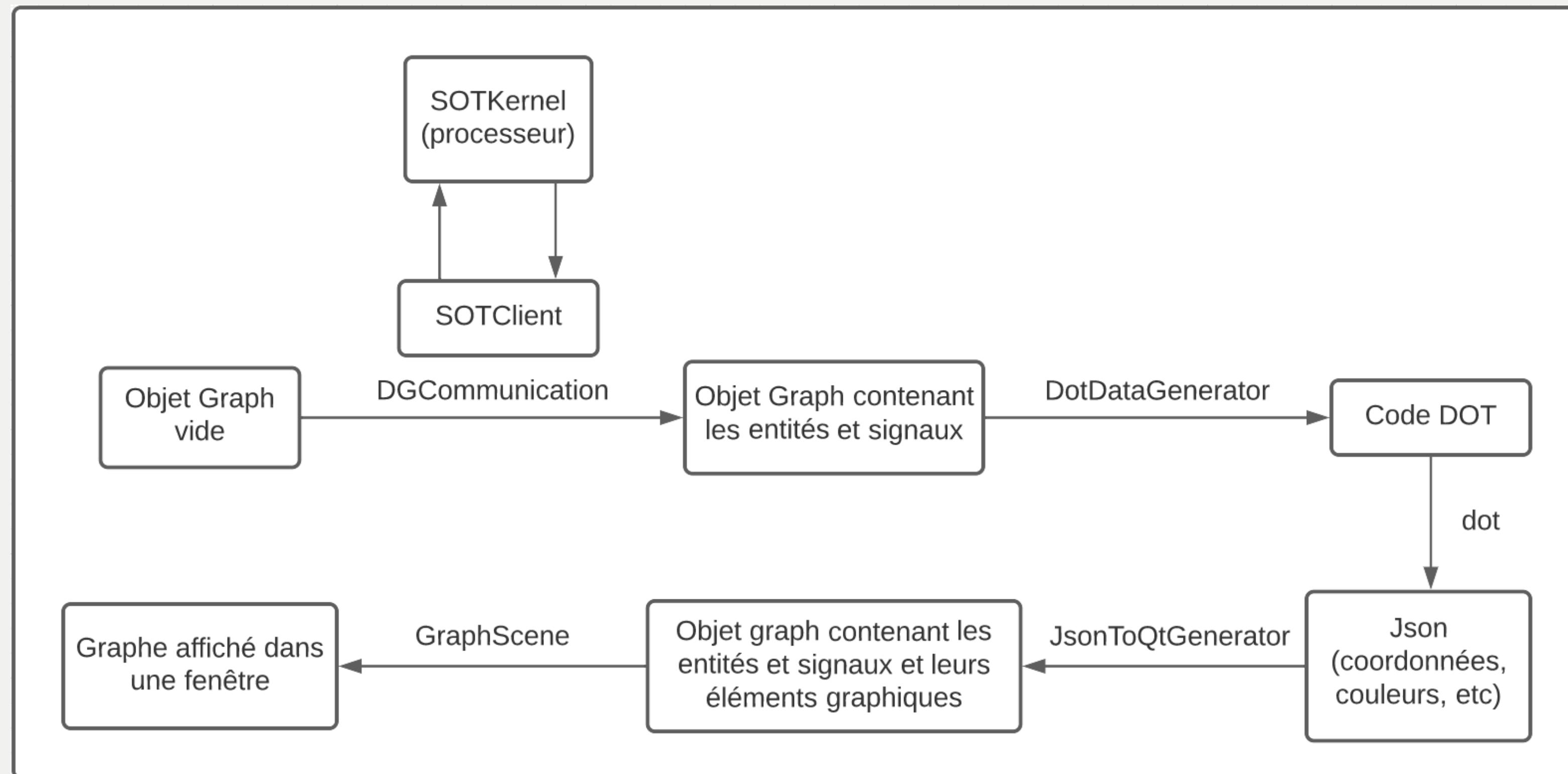
digraph G {
    subgraph cluster_0 {
        style=filled;
        color=lightgrey;
        node [style=filled,color=white];
        a0 -> a1 -> a2 -> a3;
        label = "process #1";
    }
    subgraph cluster_1 {
        node [style=filled];
        b0 -> b1 -> b2 -> b3;
        label = "process #2";
        color=blue
    }
    start -> a0;
    start -> b0;
    a1 -> b3;
    b2 -> a3;
    a3 -> a0;
    a3 -> end;
    b3 -> end;

    start [shape=Mdiamond];
    end [shape=Msquare];
}

```



Code DOT et le graphe généré



Structure globale de sot-gui