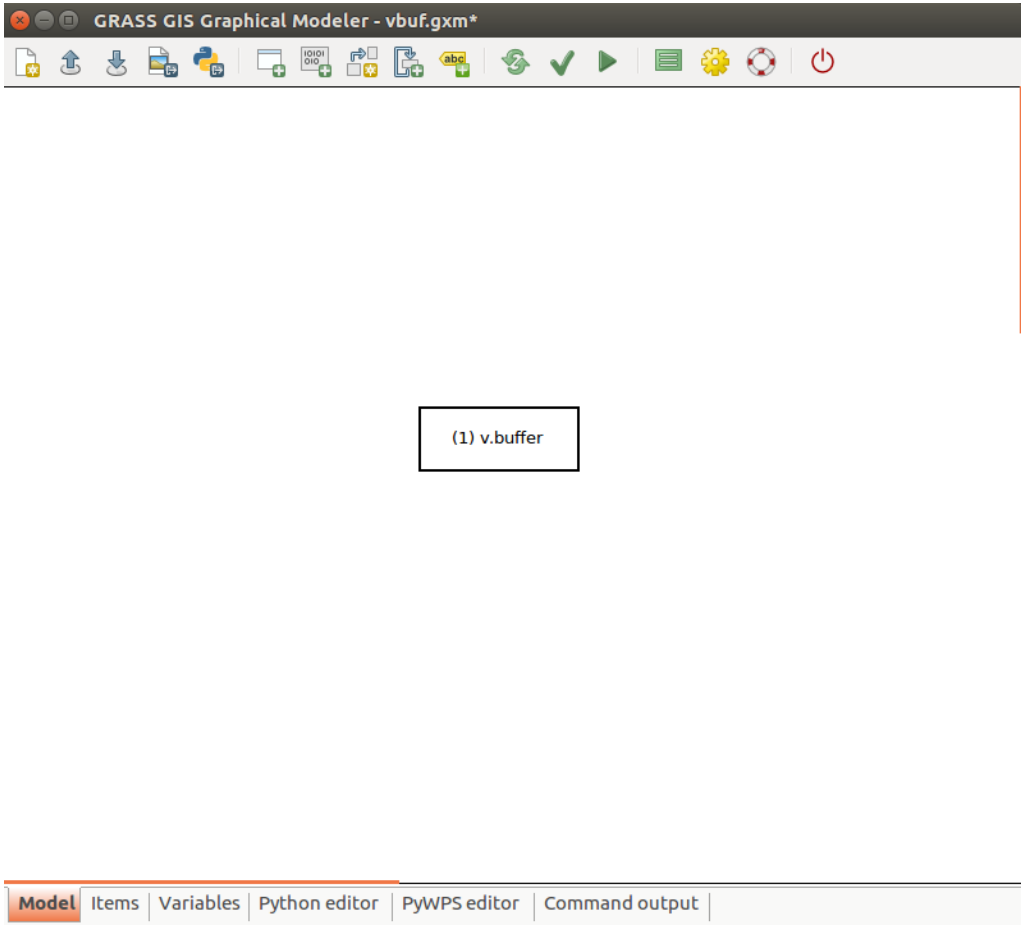
# Rozšíření GRASS modeláře

Ondřej Pešek

# Podpora uživatelského rozhraní (python export)



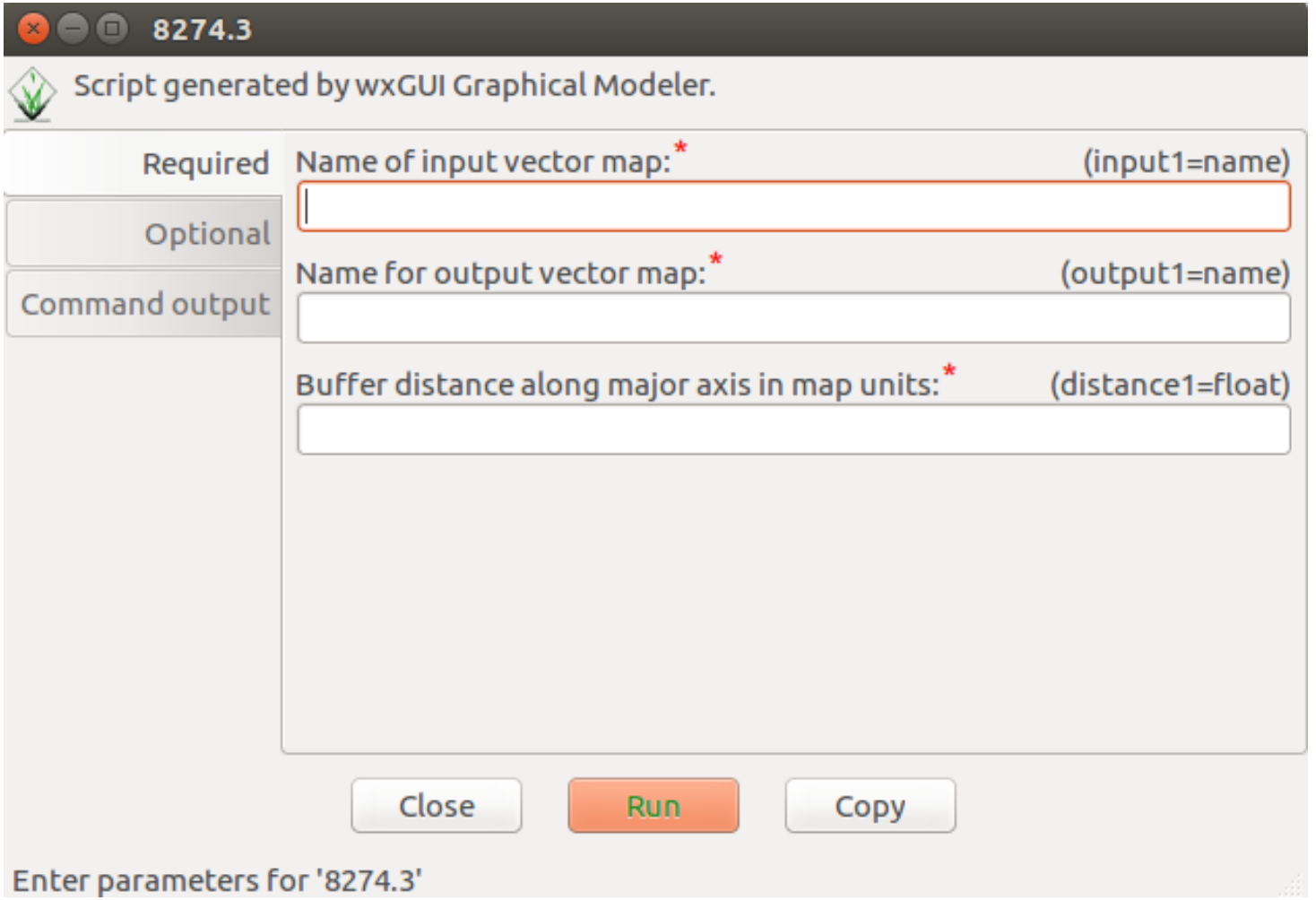
# Podpora uživatelského rozhraní (python export)

```
#%module
#% description: Script generated by wxGUI Graphical Modeler.
#%end
#%flag
#% key: t1
#% description: Transfer categories and attributes
#%end
#%option
#% key: input1
#% description: Name of input vector map
#% required: yes
#% type: string
#% key_desc: name
#%end
#%option
#% key: output1
#% description: Name for output vector map
#% required: yes
#% type: string
#% key_desc: name
#%end
#%option
#% key: distance1
#% description: Buffer distance along major axis in map units
#% required: yes
#% type: double
#%end
```

# Podpora uživatelského rozhraní (python export)

```
def cleanup():
    pass

def main(options, flags):
    run_command("v.buffer",
                flags = 's' + getParameterizedFlags(flags, ["t1"]),
                overwrite = True,
                input = options["input1"],
                layer = "-1",
                type = "point,line,area",
                output = options["output1"],
                distance = options["distance1"],
                angle = 0,
                scale = 1.0,
                tolerance = 0.01)


    return 0

def getParameterizedFlags(flags, itemFlags):
    fl = ''
    for i in [key for key, value in flags.iteritems() if value == True]:
        if i in itemFlags:
            fl += i
    return fl

if __name__ == "__main__":
    options, flags = parser()
    atexit.register(cleanup)
    sys.exit(main(options, flags))
```

# Podpora uživatelského rozhraní (python export)



# PyWPS export

PyWPS script

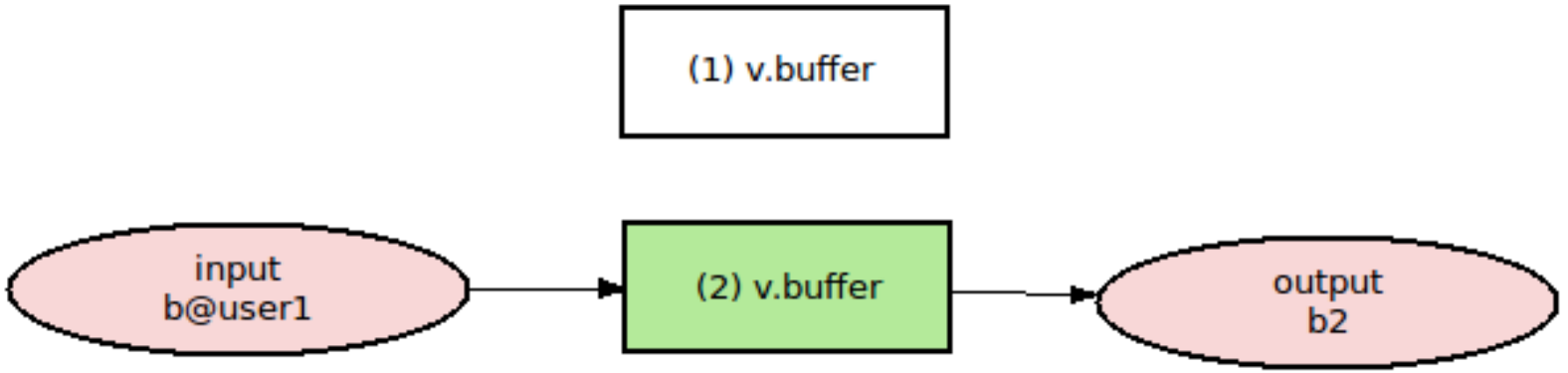
```python
#!/usr/bin/env python3

import sys
import os
import atexit
from grass.script import parser, run_command
from pywps import Process, LiteralInput, ComplexInput, ComplexOutput
from grass.pygrass.modules import Module
from pywps.app.Service import Service

class Model(Process):
    def __init__(self):
        inputs = list()
        outputs = list()

        inputs.append(LiteralInput(identifier="tl",
                                   title="Transfer categories and attributes"
        inputs.append(ComplexInput(identifier="input1",
                                   title="Name of input vector map"))
        outputs.append(ComplexOutput(identifier="output1",
                                     title="Name for output vector map"))
        inputs.append(LiteralInput(identifier="distance1",
                                   title="Buffer distance along major axis in map units",
                                   data_type="float"))

        super(Model, self).__init__(
            self._handler,
            identifier='model',
            title='model',
            inputs=inputs,
            outputs=outputs,
            abstract='Script generated by wxGUI Graphical Modeler.',
            version='1.0',
            store_supported=True,
            status_supported=True)

    @staticmethod
    def _handler(request, response):
        Module("v.buffer",
                    flags = 's' + getParameterizedFlags(flags, ["tl"]),
                    overwrite = True,
                    input=request.inputs["input1"][0].file,
                    layer = "-1",
                    type = "point,line,area",
                    output="output1",
                    distance=request.inputs["distance1"][0].file,
                    angle = 0,
                    scale = 1.0,
                    tolerance = 0.01)

        response.outputs["output1"].file = "output1"

        return response

if __name__ == "__main__":
```

# Funguje i pro více modulů?

# Funguje i pro více modulů?

ANO!

# Funguje i pro více modulů?



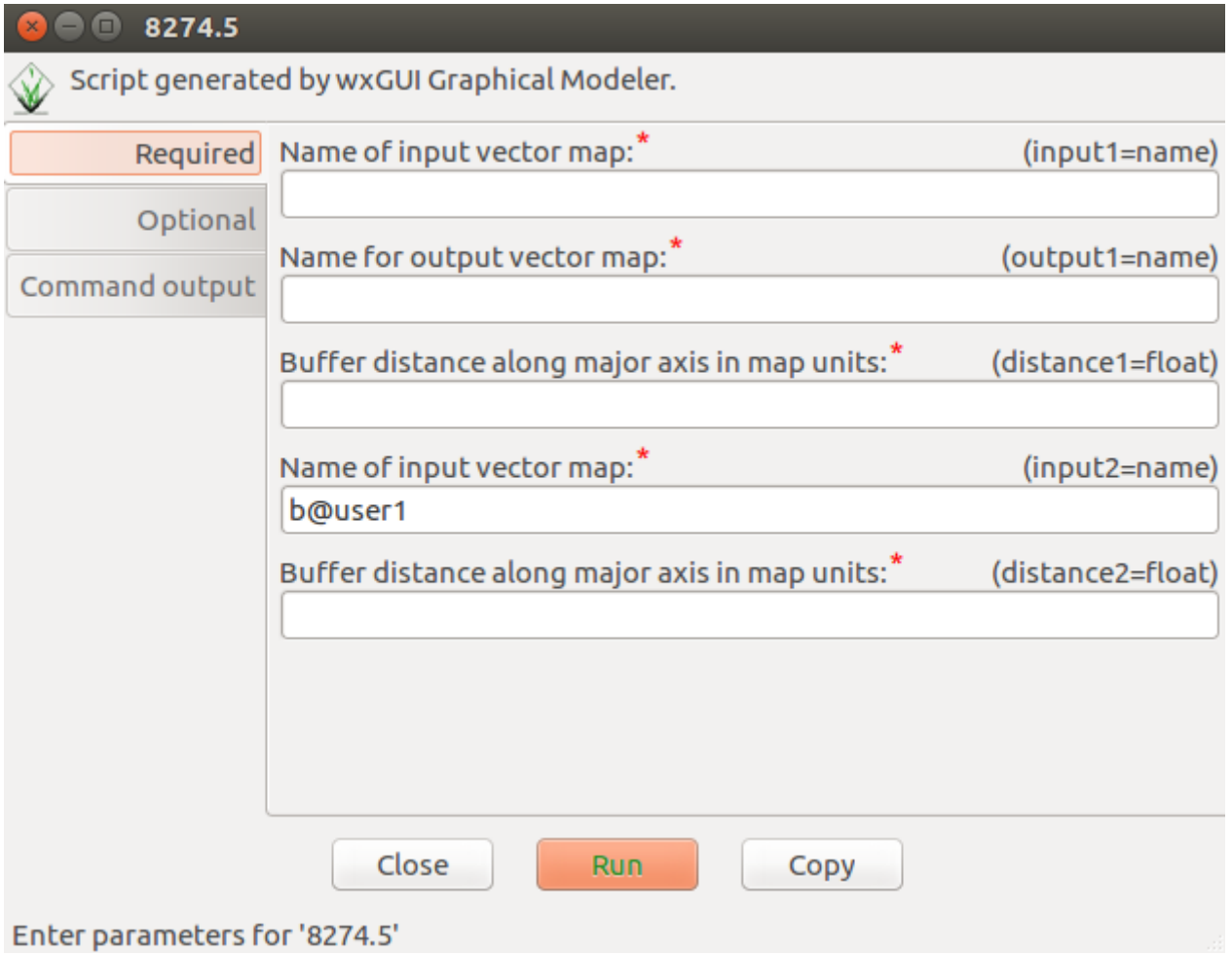
# Funguje i pro více modulů?

```
#%flag
#% key: t1
#% description: Transfer categories and attributes
#%end
#%option
#% key: input1
#% description: Name of input vector map
#% required: yes
#% type: string
#% key_desc: name
#%end
#%option
#% key: output1
#% description: Name for output vector map
#% required: yes
#% type: string
#% key_desc: name
#%end
#%option
#% key: distance1
#% description: Buffer distance along major axis in map units
#% required: yes
#% type: double
#%end
#%option
#% key: input2
#% description: Name of input vector map
#% required: yes
#% type: string
#% key_desc: name
#% answer: b@user1
#%end
#%option
#% key: distance2
#% description: Buffer distance along major axis in map units
#% required: yes
#% type: double
#%end
```

# Funguje i pro více modulů?

```
def main(options, flags):
    run_command("v.buffer",
                flags = 's' + getParameterizedFlags(flags, ["t1"]),
                overwrite = True,
                input = options["input1"],
                layer = "-1",
                type = "point,line,area",
                output = options["output1"],
                distance = options["distance1"],
                angle = 0,
                scale = 1.0,
                tolerance = 0.01)

    run_command("v.buffer",
                input = options["input2"],
                layer = "-1",
                type = "point,line,area",
                output = "b2",
                distance = options["distance2"],
                angle = 0,
                scale = 1.0,
                tolerance = 0.01)


    return 0

def getParameterizedFlags(flags, itemFlags):
    fl = ''
    for i in [key for key, value in flags.iteritems() if value == True]:
        if i in itemFlags:
            fl += i
    return fl
```

# Funguje i pro více modulů?



```
class Model(Process):
    def __init__(self):
        inputs = list()
        outputs = list()

        inputs.append(ComplexInput(identifier="input1",
            title="Name of input vector map"))
        outputs.append(ComplexOutput(identifier="output1",
            title="Name for output vector map"))
        inputs.append(LiteralInput(identifier="distance1",
            title="Buffer distance along major axis in map units",
            data_type="float"))
        inputs.append(ComplexInput(identifier="input2",
            title="Name of input vector map"))
        inputs.append(LiteralInput(identifier="distance2",
            title="Buffer distance along major axis in map units",
            data_type="float"))

        super(Model, self).__init__(
            self._handler,
            identifier='model',
            title='model',
            inputs=inputs,
            outputs=outputs,
            abstract='Script generated by wxGUI Graphical Modeler.',
            version='1.0',
            store_supported=True,
            status_supported=True)

    @staticmethod
    def _handler(request, response):
        Module("v.buffer",
                flags = 's' + getParameterizedFlags(flags, ["t1"]),
                overwrite = True,
                input=request.inputs["input1"][0].file,
                layer = "-1",
                type = "point,line,area",
                output="output1",
                distance=request.inputs["distance1"][0].file,
                angle = 0,
                scale = 1.0,
                tolerance = 0.01)
        Module("v.buffer",
                input=request.inputs["input2"][0].file,
                layer = "-1",
                type = "point,line,area",
                output = "b2",
                distance=request.inputs["distance2"][0].file,
                angle = 0,
                scale = 1.0,
                tolerance = 0.01)

        response.outputs["output1"].file = "output1"

        return response
```

# Děkuji za pozornost

*„Neexistují hloupé otázky, pouze hloupé odpovědi.“*

*- Aleš Čepek*

*- 11. 11. 2014, 8. 3. 2015, 25. 10. 2015, 17. 12. 2015, 5. 10. 2016, 12. 10. 2016, 5. 11. 2016, 12. 12. 2016, 5. 1. 2017*