

## Lab 9A: YANG (Yet Another Next Generation)

### Install and run pyang on a Raspberry Pi

```
pi@Zhaoning: ~/demo
File Edit Tabs Help
</notification>
</module>
pi@Zhaoning:~/demo $ pyang -f uml -o intrusiondetection.uml intrusiondetection.yang --uml-no=stereotypes,annotation,typedef
pi@Zhaoning:~/demo $ cat intrusiondetection.uml
'Download plantuml from http://plantuml.sourceforge.net/
'Generate png with java -jar plantuml.jar <file>
'Output in img/<module>.png
'If Java spits out memory error increase heap size with java -Xmx1024m -jar plantuml.jar <file>
@startuml img/intrusiondetection.png
hide empty fields
hide empty methods
hide <<case>> circle
hide <<augment>> circle
hide <<choice>> circle
hide <<leafref>> stereotype
hide <<leafref>> circle
hide stereotypes
page 1x1
Title intrusiondetection
package "intrusion:intrusiondetection" as intrusion_intrusiondetection {
class "intrusiondetection" as intrusiondetection << (N, #33CCFF) module>>
class "room" as intrusiondetection_I_room_grouping << (G, lime) grouping>>
intrusiondetection_I_room_grouping : doorsensorID : string
intrusiondetection_I_room_grouping : motionsensorID : string
class "intrusiondetection" as intrusiondetection_I_intrusiondetection <<container>>
intrusiondetection_I_intrusiondetection -- "0..1" intrusiondetection_I_intrusiondetection
intrusiondetection_I_intrusiondetection : systemID : string {mandatory} {Config : false}
intrusiondetection_I_intrusiondetection : systemLocation : string {mandatory} {Config : false}
intrusiondetection_I_intrusiondetection : systemStatus : enumeration : {up,down,armed,...} {mandatory} {Config : false}
class "sensors" as intrusiondetection_I_intrusiondetection_I_sensors <<container>>
intrusiondetection_I_intrusiondetection_I_sensors -- "1" intrusiondetection_I_intrusiondetection_I_sensors
intrusiondetection_I_intrusiondetection_I_sensors : room {uses}
intrusiondetection : arm-system()
intrusiondetection : disarm-system()
class "systemArmed" as intrusiondetection_I_systemArmed << (N, #00D182) notification>>
intrusiondetection -- intrusiondetection_I_systemArmed : notification
intrusiondetection_I_systemArmed : armStatus : enumeration : {armed,disarmed,error,}
}
intrusiondetection_I_intrusiondetection_I_sensors --> intrusiondetection_I_room_grouping : uses
center footer
<size:20> UML Generated : 2020-11-15 17:13 </size>
endfooter
@enduml
pi@Zhaoning:~/demo $
```

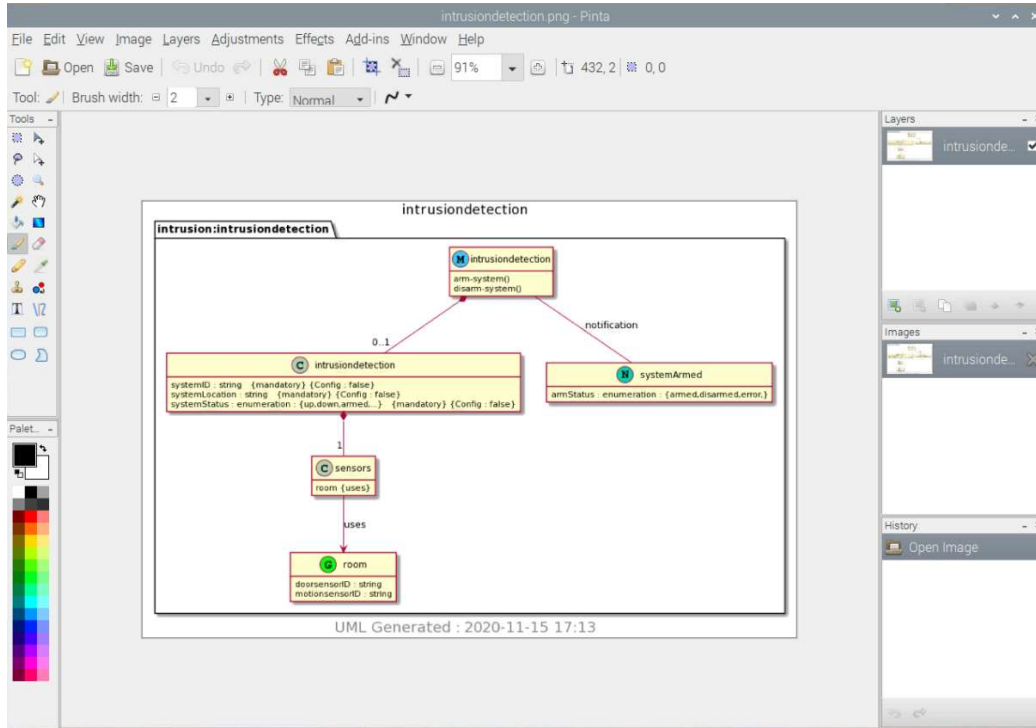
### Install PlantUML on a laptop or Raspberry Pi

```
pi@Zhaoning:~/demo $ sudo pip3 install -U plantuml
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting plantuml
  Downloading https://files.pythonhosted.org/packages/a9/1a/4603314acf466fdad91b7f6c83eb1364a7e279f9a8805febe3554f17faf6/plantuml-0.3.0-py3-none-any.whl
Requirement already satisfied, skipping upgrade: http://lib2 in /usr/local/lib/python3.7/dist-packages (from plantuml) (0.18.1)
Installing collected packages: plantuml
Successfully installed plantuml-0.3.0
pi@Zhaoning:~/demo $
```

### Run PlantUML to create a sequence diagram in PNG

```
Successfully installed plantuml-0.3.0
pi@Zhaoning:~/demo $ python3 -m plantuml intrusiondetection.uml
[[{"filename": "intrusiondetection.uml", "gen_success": True}]]
pi@Zhaoning:~/demo $
```

# Install and run GIMP and Pinta to display a PNG file via VNC Viewer



```
pi@Zhaoning: ~/demo
File Edit Tabs Help

126 new root certificates were added to your trust store.
Import process completed.
Done
done.
pi@Zhaoning:~ $ cd ~/demo
pi@Zhaoning:~/demo $ pinta intrusiondetection.png
pi@Zhaoning:~/demo $ gimp -h
Usage:
  gimp [OPTION...] [FILE|URI...]

GNU Image Manipulation Program

Help Options:
  -h, --help                Show help options
  --help-all               Show all help options
  --help-gegl               Show GEGL Options
  --help-gtk                Show GTK+ Options

Application Options:
  -v, --version             Show version information and exit
  --license                 Show license information and exit
  --verbose                 Be more verbose
  -n, --new-instance        Start a new GIMP instance
  -a, --as-new              Open images as new
  -i, --no-interface        Run without a user interface
  -d, --no-data             Do not load brushes, gradients, patterns, ...
  -f, --no-fonts            Do not load any fonts
  -s, --no-splash           Do not show a splash screen
  --no-shm                  Do not use shared memory between GIMP and plug-ins
  --no-cpu-accel             Do not use special CPU acceleration functions
  --session=<name>          Use an alternate sessionrc file
  -g, --gimprc=<filename>   Use an alternate user gimprc file
  --system-gimprc=<filename> Use an alternate system gimprc file
  -b, --batch=<command>     Batch command to run (can be used multiple times)
  --batch-interpreter=<proc> The procedure to process batch commands with
  -c, --console-messages    Send messages to console instead of using a dialog
  --pdb-compat-mode=<mode>  PDB compatibility mode (off|on|warn)
  --stack-trace-mode=<mode> Debug in case of a crash (never|query|always)
  --debug-handlers           Enable non-fatal debugging signal handlers
  -g-fatal-warnings          Make all warnings fatal
  --dump-gimprc             Output a gimprc file with default settings
  --show-playground         Show a preferences page with experimental features
  --display=DISPLAY         X display to use

pi@Zhaoning:~/demo $
```

