

# Pogo Status update

---

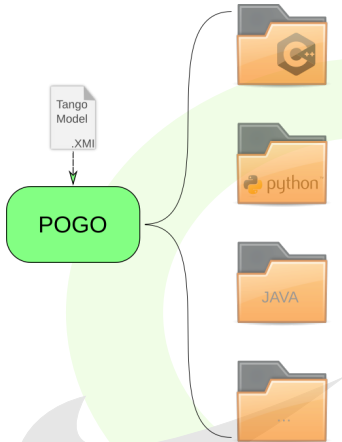
Damien Lacoste

May 29 2024

controls



# POGO - Program Obviously used to Generate tango Objects



- ▶ Tango code generator written in java swing.
- ▶ Use a xmi file to describe tango class and device server model.
- ▶ Support device servers with several classes.
- ▶ GUI to create and manage tango class and device server model.
- ▶ Support generators for different languages and uses:
  - ▶ C++ code generator.
  - ▶ java code generator.
  - ▶ Python code generator, in 2 flavors.
  - ▶ HTML documentation generator.
- ▶ Run with java 11+.



# Pogo roadmap update

## Current status

What's brewing?

What's missing?

*controls*



# Roadmap check

- ▶ Java 17 migration. ✓
- ▶ Consolidate CI/CD. ✓
- ▶ Python rewrite. ✓
- ▶ C++ generator code reorganization. ✓
- ▶ Move away from protected regions. ✓

controls



# Version support

- ▶ Pogo 9.6
  - ▶ Last version with java 8 support
  - ▶ Only support cpptango 9.3.x
  - ▶ Not supported anymore
- ▶ Pogo 9.7
  - ▶ Java 11 to 15 support.
  - ▶ Last version with cpptango 9.3.x support
  - ▶ Support till the end of LTS in cpptango
- ▶ Pogo 9.8
  - ▶ Java 11 to 15 support.
  - ▶ cpptango 9.4.x and onward support
  - ▶ Minimal support, try to move to newer version.



- ▶ Pogo 9.9
  - ▶ Java 17 support.
  - ▶ cpptango 9.4.x and onward support
  - ▶ Stable version.
- ▶ Pogo 9.10
  - ▶ Last java version.
  - ▶ cpptango 9.4.x and onward support.
  - ▶ LTS for as long as needed.
- ▶ Pogo 10
  - ▶ Python rewrite.
  - ▶ Future stable.



# Pogo roadmap update

Current status

**What's brewing?**

What's missing?

*controls*



## Pogo 9.10: C++ code generator refactoring

```
± tree
.
├── classes
│   └── Test
│       ├── CMakeLists.txt
│       ├── include
│       │   ├── TestClass.h
│       │   └── Test.h
│       └── src
│           ├── TestClass.cpp
│           ├── Test.cpp
│           └── TestStateMachine.cpp
├── CMakeLists.txt
├── doc
├── servers
│   └── Test
│       ├── CMakeLists.txt
│       └── src
│           ├── ClassFactory.cpp
│           └── main.cpp
├── test
└── Test.xmi

9 directories, 11 files
```

- ▶ Clear separation between headers and source files.
- ▶ We can see the tango model structure of the classes and device servers.
- ▶ All projects are managed the same, whatever the number of classes they ship.
- ▶ Modern cmake integration with targets, and support for cmake installation.
- ▶ Full control about what to build, may it be a library, binaries, linking to static or shared library, activating support for part of the classes or not through the use of consistent cmake flags.



## Pogo 9.10: Transition tool

- ▶ Will support legacy code generation and C++ refactoring.
- ▶ Will support legacy code generation and to a certain level code generation without protected regions.
- ▶ Bug fixes, dependencies update.
- ▶ Minimal or no support for new features.



## Python rewrite

- ▶ Use templates with jinja2.
- ▶ modular for the outputs and the inputs.
- ▶ support xmi as well as yaml and json.
- ▶ You never have to edit generated code.
- ▶ Generate a simulated device to test.

## Old java implementation

- ▶ Rely on xtext/xtend, which is a niche framework.
- ▶ Monolithic hard to maintain app.
- ▶ Can only work with xmi.
- ▶ Rely on protected regions.



## Python rewrite

- ▶ Execution time average for 100 runs.
  - ▶ 0.38s
- ▶ Lines of Code.
  - ▶ about 7000
- ▶ Github's stars for jinja2.
  - ▶ 9.3K

## Old java implementation

- ▶ Execution time average for 100 runs.
  - ▶ 2.98s
- ▶ Lines of Code.
  - ▶ about 130000
- ▶ Github's stars for xtext.
  - ▶ 703
- ▶ Protected regions are a bad pattern!

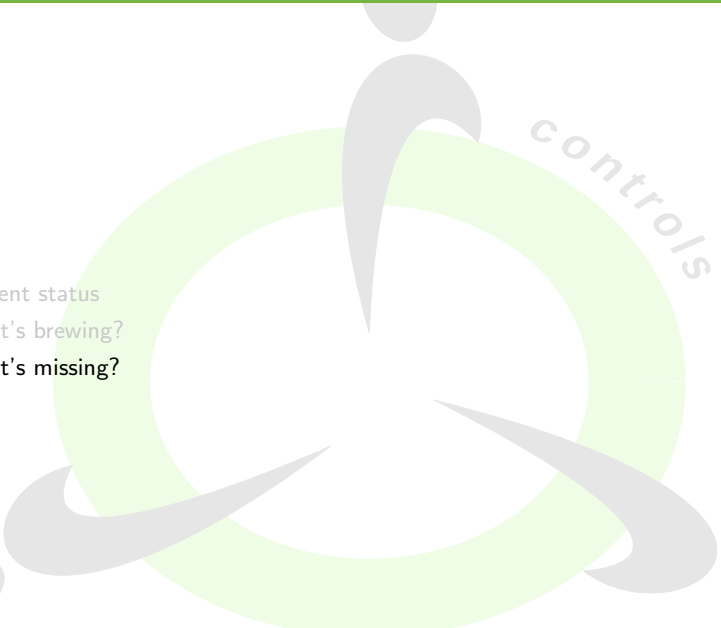


## Check the repo

- ▶ Specifications for the input format.
- ▶ C++ template more or less ready, without protected regions.
- ▶ Python template available for pythonHL only
- ▶ Java template available



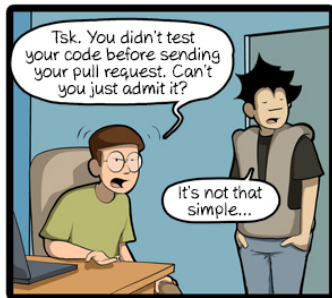
# Pogo roadmap update



Current status  
What's brewing?  
**What's missing?**

controls





CommitStrip.com



## But also...

- ▶ Tests.
- ▶ Legacy python template.
- ▶ Support for linters.
- ▶ To be ready on time for the next cppTango release and DevDict.
- ▶ Should we rename the project?



# POGO



AMFORD  
LLOYD





Any Questions ?  
Thanks !





Documentation

Project repository

<https://isocpp.org>

<https://www.python.org>

Jinja2 documentation