Extending and embedding the Python interpreter

Michael Wegner

2018-04-24

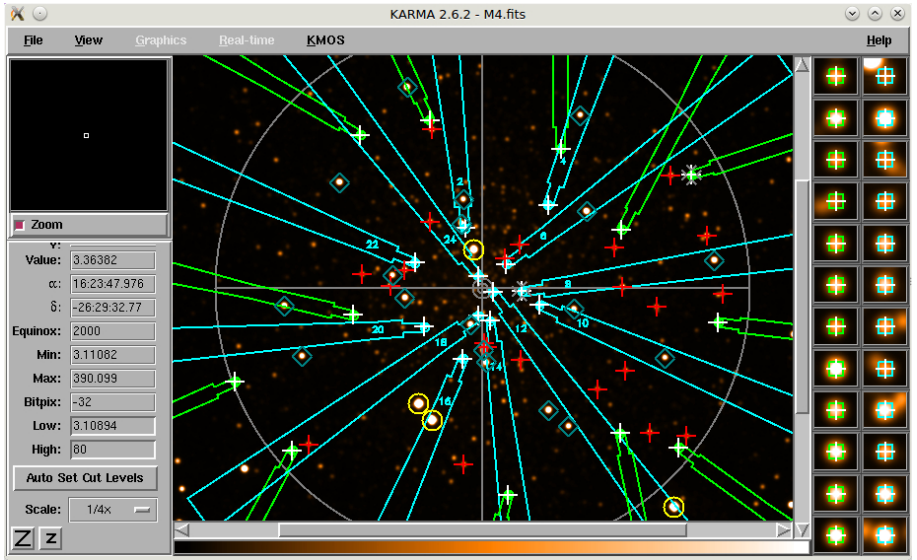
Sometimes neither Python nor C/C++ alone is sufficient.

- For performance reasons some parts of a scripting language (e.g. Python) application are better implemented in C/C++.
- Third party libraries written in C/C++ shall be used/incorporated in a Python program.
- An application shall be script-driven but not implemented in Python as a whole.

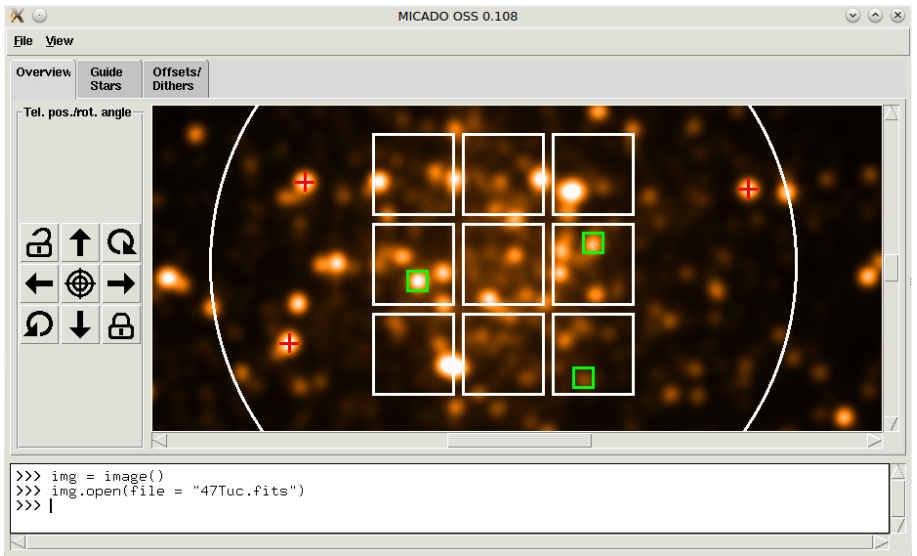
⇒ Interface/API between scripting language and C/C++ required.

Typical situation for observation preparation tools.

KMOS preparation tool: Tcl/Tk/C++



MICADO preparation tool: C++/Python



The C API enables almost everything to be implemented.

The CPython API

- provides a comprehensive set of C functions that give access to the Python interpreter in every aspect.
- is usable from C as well as from C++.

With the CPython API you can

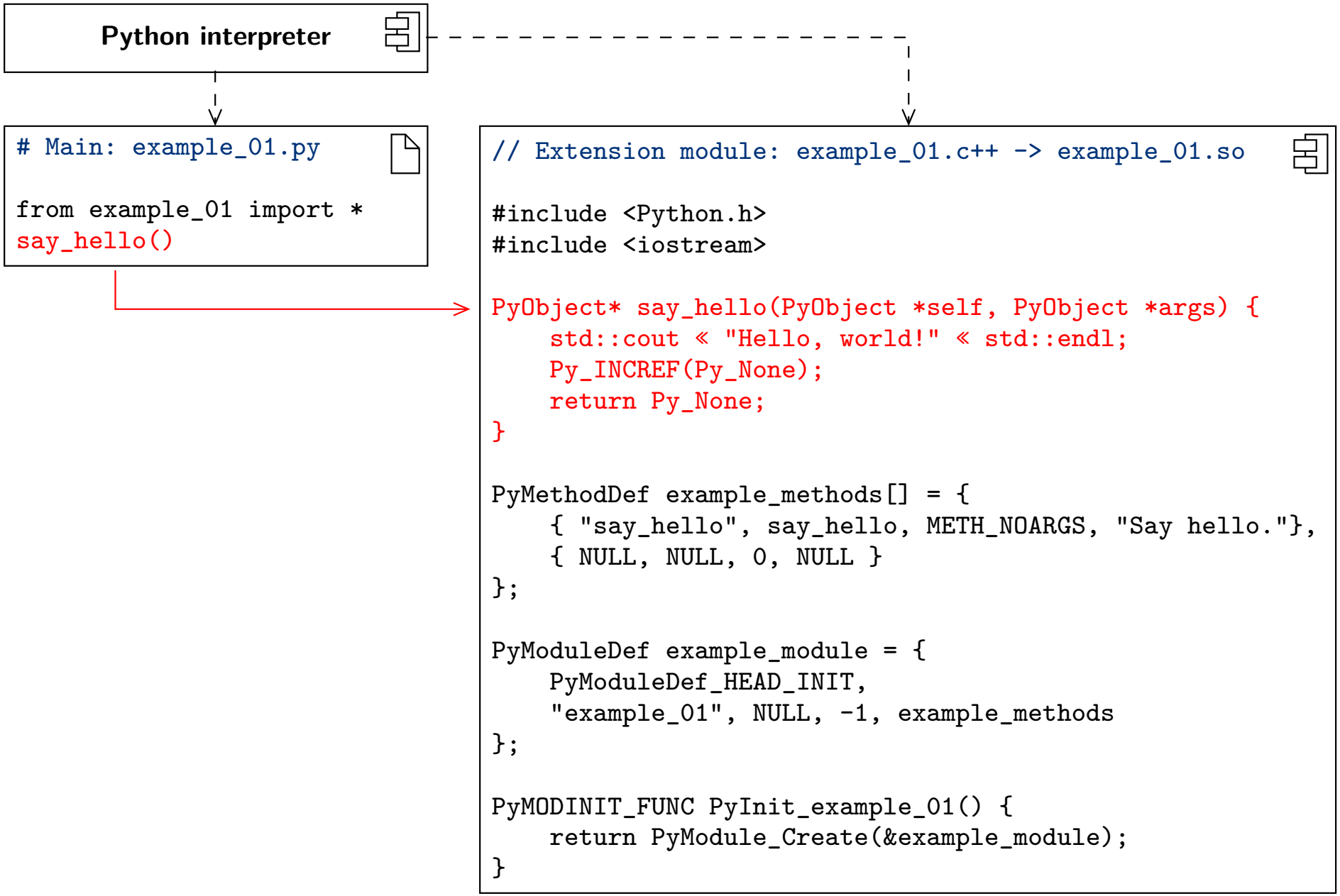
- write *extension modules*.
- *embed the Python interpreter* in a custom application.

A number of tools may facilitate things:

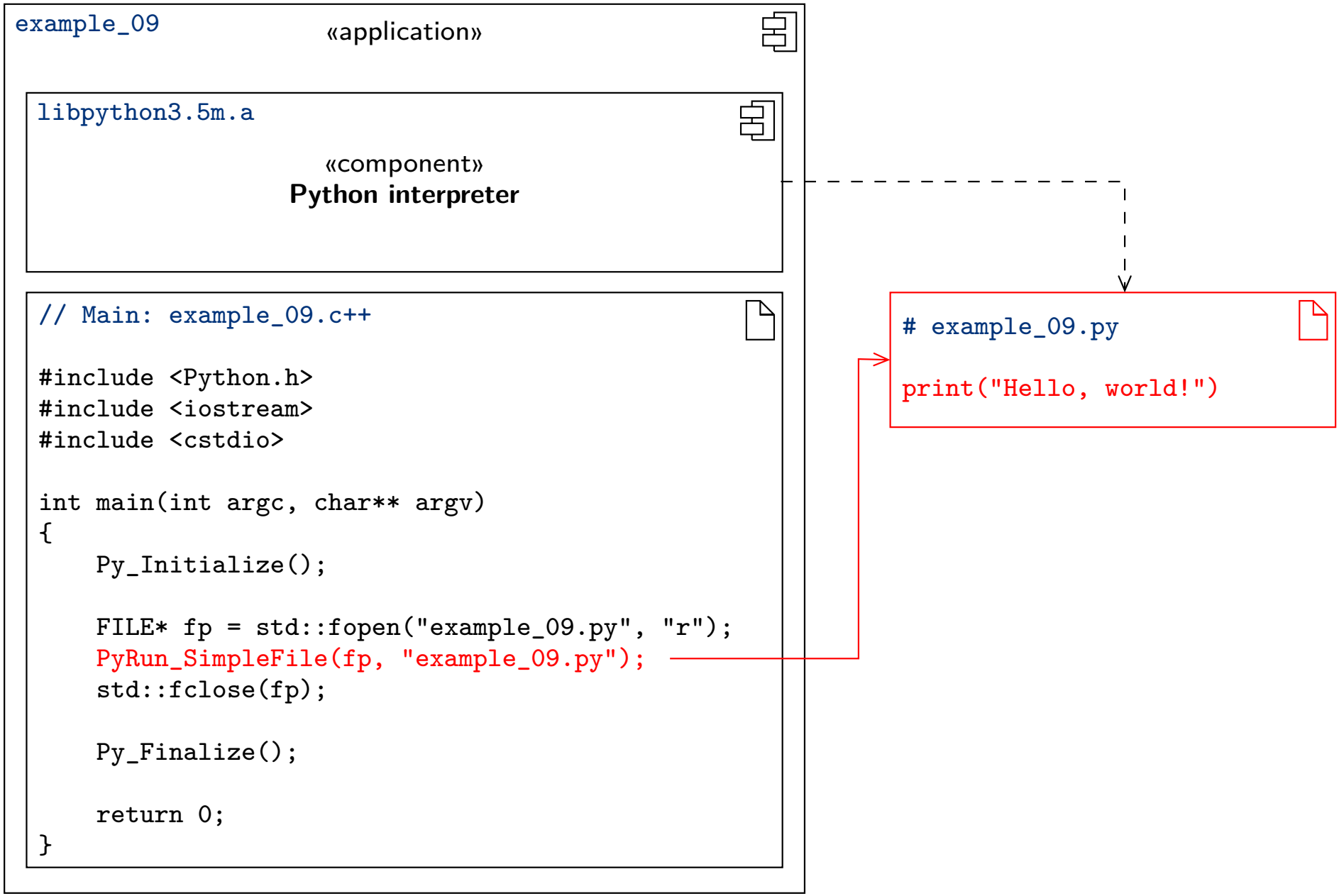
distutils	Native Python tool that facilitates cross-platform build and distribution of extension modules.
SWIG	**S**implified **W**rapper and **I**nterface **G**enerator that connects programs written in C and C++ with scripting languages.
boost::python	C++ library which enables interoperability between C++ and Python, particularly for exposing existing C++ classes to Python.
Cython	Superset of Python language that generates and compiles C code.

Using the low-level functions directly gives maximum flexibility.

Extension modules provide new functionality as shared libs.



An embedded interpreter is part of a custom application.



Tutorials are available online and in print.

Python online documentation:
`https://docs.python.org/3/extending/index.html`

Another tutorial:
`http://www.tutorialspoint.com/python/python_further_extensions.htm`

And yet another one:
`https://en.wikibooks.org/wiki/Python_Programming/Extending_with_C`

Lutz, M.:

Programming Python, 4th edition.

O'Reilly, 2011.

Example source code from this talk:
`www.usm.lmu.de/people/wegner/seminar/2018-04-24/example.tar.gz`