



Object Oriented Scripting Language
Pythonの
KEKB制御システムへの応用

高エネルギー物理学におけるオブジェクト指向技術の応用研究会

1998, March 11

KEK加速器研究施設
山本昇



目次

KEKB制御システム

Python

Python in KECBコントロールシステム



KEKB制御システム

EPICS (Experimental Physics and Industrial Control System)

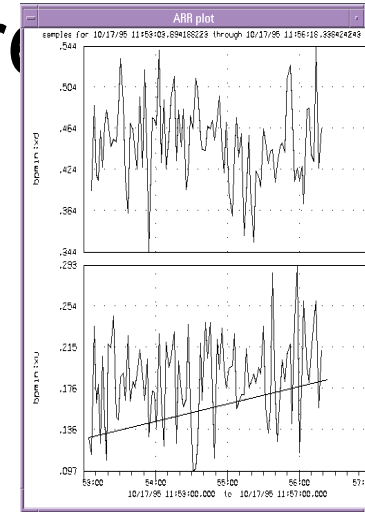
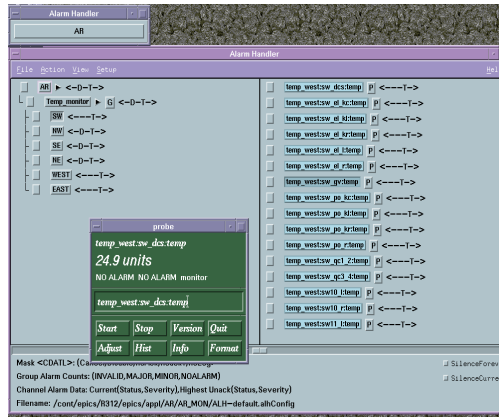
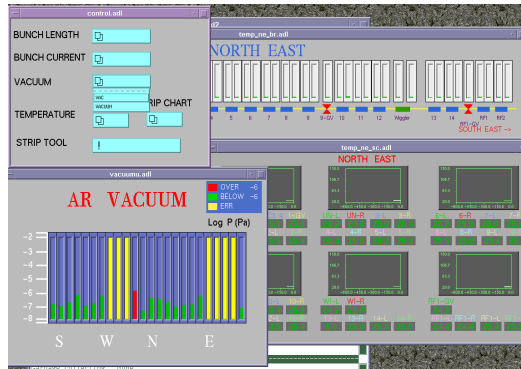
- ◆ Runtime Database
- ◆ Software Toolkit
- ◆ Channel Access (Network protocol over TCP/IP)

EPICS Database Record

- ◆ Distributed
- ◆ Defined Action(Hardware Access etc.)



EPICS Architecture



MEDM
+
.adl ファイル

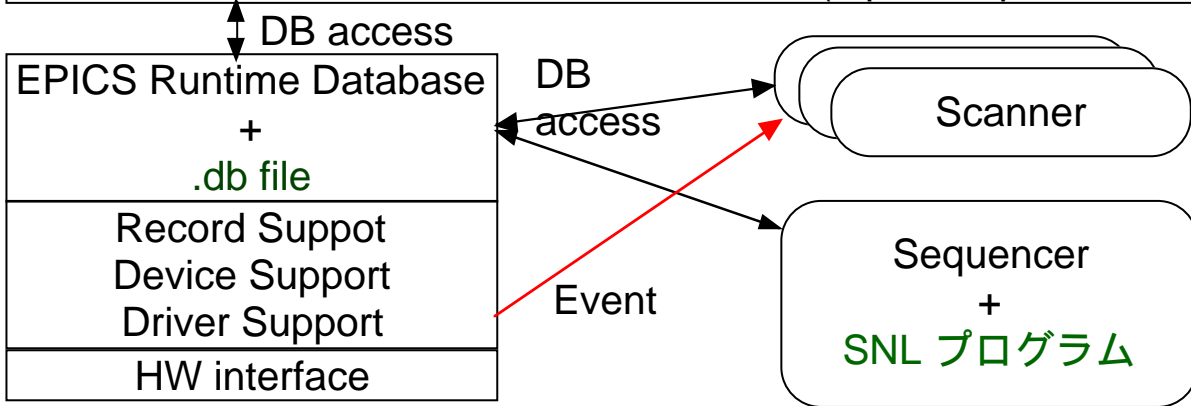
ALH
+
.alhConfig

Archiver
+
request

Channel Access Library Interface

TCP/IP & UDP/IP

Channel Access Server on IOC (Input Output Controller)





Two Language system

実用的なシステムでは2種類の言語が必要

- ◆ ベーシックなオブジェクト作成言語
- ◆ オブジェクトを組み合わせるスクリプト言語

C and Shell in UNIX

PCL and Nodal in TRISTAN control system

Java and JavaScript

C/C++/SNL and Python in KEKB Control System



Python

What is Python

- ◆ Object Oriented
- ◆ Scripting
- ◆ Multi Platform
- ◆ Public Domain
- ◆ Language



Python

What is Python

- ◆ Object Oriented
 - Class and its method
 - Multiple inheritance
 - support Tk-Widget as objects
- ◆ Scripting
- ◆ Multi Platform
- ◆ Public Domain
- ◆ Language



Python

What is Python

- ◆ Object Oriented
- ◆ Scripting
 - No compilation
 - Rapid development cycle
- ◆ Multi Platform
- ◆ Public Domain
- ◆ Language



Python

What is Python

- ◆ Object Oriented
- ◆ Scripting
- ◆ Multi Platform
 - Mac, Unix, Windows
- ◆ Public Domain
- ◆ Language



Python

What is Python

- ◆ Object Oriented
- ◆ Scripting
- ◆ Multi Platform
- ◆ Public Domain
 - A source code is available for free.
- ◆ Language



Python

What is Python

- ◆ Object Oriented
- ◆ Scripting
- ◆ Multi Platform
- ◆ Public Domain

- ◆ Language
 - ▶ Clean Syntax
 - ▶ Easy to maintain
 - ▶ Extendible
 - Modules in C/C++/(Java),...
 - Embedded in other applications.



Python and other Scripting Languages

	Python/Tk	Tcl/TK	PERL/TK	Java
言語タイプ	インタプリタ (中間コード自動生成)	インタプリタ (Scripting Language)	インタプリタ (Scripting Language)	Java VM語へのコンパイラ
プラットフォーム	Mac/Win/Unix	Mac/Win/Unix	Win/Unix	Mac/Win/Unix
オブジェクト指向	Class/多重継承/ module	?	?	Pure OOPL Ju
開発環境	Any Text Editor	Any Text Editor	Any Text Editor	Any Text Editor+JDK
配付形態	FreeWare Source code	SUN/Free ware Source code	FreeWare Source code	SUN proprietary Development tool/VM are commercially available
Interface Builder	SpecPython	SpecTcl/XF	?	Commercial products
その他	玄人好み (?)	Loosing its power	Used as a build tool in EPICS CGI	Possible Incompatibility between Java VMs. HOT!



Python Example

```
class Hello:
    def __init__(self):
        print "Welcome to the WORLD"
    def __del__(self):
        print "Good bye WROLD"
    def Say(self):
        print "Hello World!"
    def SayMore(self):
        print "How are you doing today?"

if(__name__ == "__main__"):
    foo=Hello()
    foo.Say()
    foo.SayMore()
    del foo
```

```
abco1.18: python Hello.py
Welcome to the WORLD
HelloWorld
How are you doing today?
Good bye WROLD
abco1.19:
```



Python Example: Tkinter

```
from Tkinter import *
```

```
class Application(Frame):
```

```
    def say_hi(self):
```

```
        print "Hello, World!"
```

```
    def createWidgets(self):
```

```
        self.QUIT=Button(self)
```

```
        self.QUIT["text"]="QUIT"
```

```
        self.QUIT["fg"]="red"
```

```
        self.QUIT["command"]=self.quit
```

```
        self.QUIT.pack({"side":"left"})
```

```
        self.hi_there=Button(self,\
```

```
        {"text":"Hello","command":self.say_hi,"fg":"blue"})
```

```
        self.hi_there.pack({"side":"left"})
```

```
    def __init__(self, master=None):
```

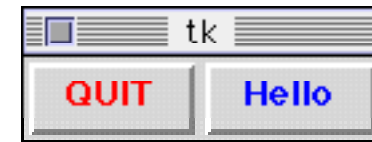
```
        Frame.__init__(self, master)
```

```
        self.pack()
```

```
        self.createWidgets()
```

```
app=Application()
```

```
app.mainloop()
```





Python in KEKB Control System

EPICS CA interface

- ◆ Channel Object
 - ▶ EPICSのCA をオブジェクト化

ORACLE Interface(oracledb module from)

- ◆ コネクションオブジェクト
- ◆ カーソルオブジェクト

Tkinter: GUI

- ◆ Tcl/Tk Library interface

Embedded in SAD(Accelerator Modeling Program)



Python in KEKB Control System:example

```
# =====  
# Main program  
# =====
```

```
f=Frame()  
f.pack(side="left")
```

```
b=Button(f.master,text="Quit")  
b.config(command=b.quit)  
b.pack(side="right",fill="y",expand="y")
```

```
CaTest("temp:ai",f).pack(side="left")  
f.mainloop()
```

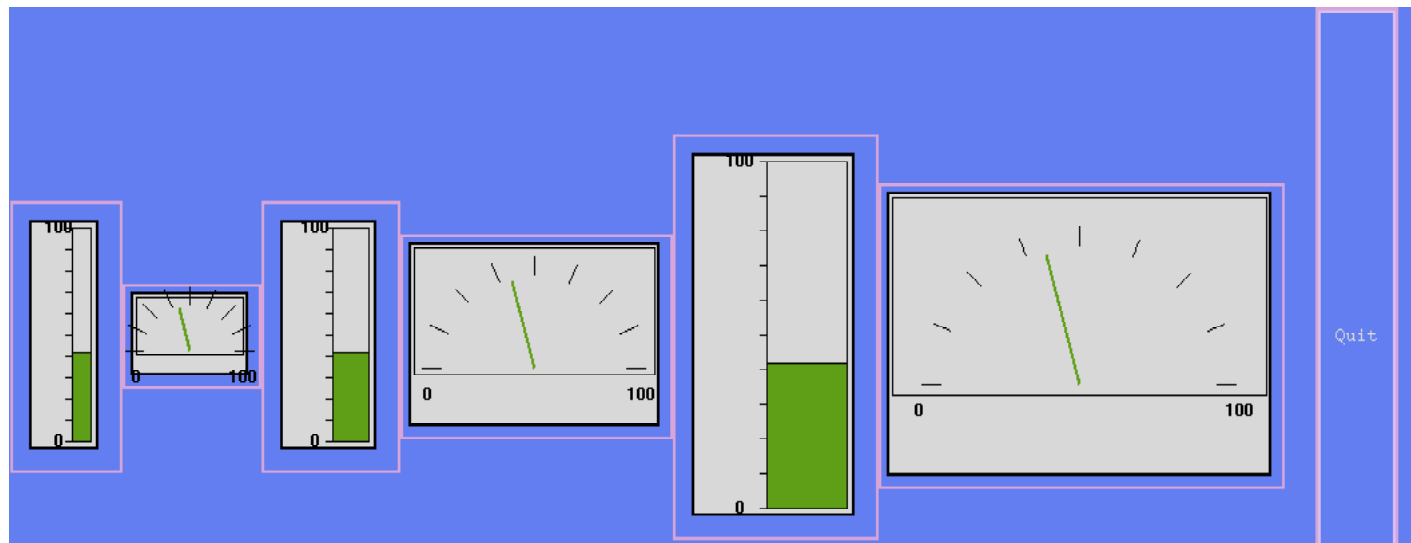
```
import ca  
import Graph  
from Tkinter import *
```

```
class MeterAndBar(ca.channel,Frame):  
    def __init__(self,name,master=None,*cnf):  
        Frame.__init__(self,master,cnf)  
        self.pack(expand=1,fill='both')  
        ca.channel.__init__(self,name);self.pendio(0.1)  
        self.get(); self.pendio()  
        self.barChart=Graph.BarChart(100, 200, 0, 100)  
        self.meter=Graph.Meter(200, 0, 100)  
        self.monitor(self.update); self.pendio()  
  
    def update(self,val):  
        self.barChart.move(val[0]); self.meter.move(val[0])
```

```
class CaTest(ca.channel,Frame):  
    def __init__(self,name,master=None,*cnf):  
        Frame.__init__(self,master)  
        self.pack(expand=1,fill='both')  
  
        self.barChart1=MeterAndBar(name,master=self)  
        self.barChart2=MeterAndBar(name,master=self)  
        self.barChart3=MeterAndBar(name,master=self)
```




Python in KEKB Control System: example output





Python in KEKB control System: ORACLE Access

```
abco1.20: python
```

```
Python 1.5 (#32, Jan 30 1998, 20:30:02) [GCC 2.7.2.2] on hp-uxB
```

```
Copyright 1991-1995 Stichting Mathematisch Centrum, Amsterdam
```

```
>>> import oracledb
```

```
>>> con=oracledb.oracledb("userid/password")
```

```
>>> cur=con.cursor()
```

```
>>> dir(cur)
```

```
['close', 'execute', 'fetchall', 'fetchmany', 'fetchone', 'setinputsizes', 'setoutputsize']
```

```
>>> cur.execute("select * from co.sad")
```

```
>>> while 1:
```

```
... try:
```

```
... d=cur.fetchone()
```

```
... print d[0],d[4],d[5],d[6]
```

```
... except:
```

```
... break
```

```
...
```



Multiple Inheritance in Python

```
>>> class adam:
...     def name(self):
...         print 'adam'
...
>>> class eve:
...     def name(self):
...         print 'eve'
...
>>> class human(adam,eve):
...     pass
...

>>> class evil(eve,adam):
...     pass
...
>>> h=human()
>>> e=evil()
>>> h.name()
adam
>>> e.name()
eve
>>> class child(adam,eve):
...     def name(self):
...         adam.name(self)
```



Summary

Python is a powerful tool to glue Objects for high level applications.

We are promoting Python as a standard tool for High level application in KEKB control system.



References

Programming Python

- ◆ Mark Lutz; O'Reilly & Associates, Inc.
- ◆ 日本語版「Python入門」オーム社より発売

Internet Programming with Python

- ◆ Aaron Watters, Guido van Rossum, Jim Ahlstrom

