

Using Python as wide-spectrum language

*experiences
in tough application domains*

Theo de Ridder (Prometa Ratum bv)

Inspiring viewpoints

- Literate Programming (Knuth)
- The Timeless Way of Building (Alexander)
- Information Visualization (Shneiderman)
- The Humane Interface (Raskin)
- Extreme Programming (Beck)

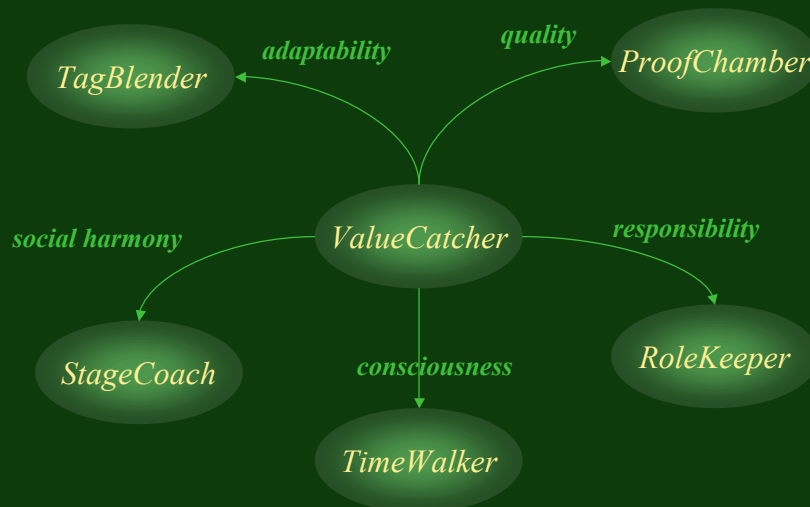


Personal keystones

- TimeWalker
multi-focal time lens
- ProofChamber
co-box testing framework
- RoleKeeper
hermetic role-based access control
- TagBlender
tagless pythonized data-hierarchies
- StageCoach
groupware as goal-oriented RPG
- ValueCatcher
generalized system analyser

3

Conceptual coherence



4

TimeWalker (1)

- features
 - unique, human(e) interface
 - scales up to huge amounts of data
 - zoomlens covers 40 years down to 50 msec
 - imports arbitrary data format
 - exports to universal and compact format
- added value
 - (over)viewing history without filters
 - visual datamining based on human pattern-recognition
 - combining arbitrary types of logdata
- requirements
 - Python 2.2
 - wxPython 2.3.2
 - Win32, Linux (Suse 8.0), Mac OSX
- status
 - beta version at sourceforge

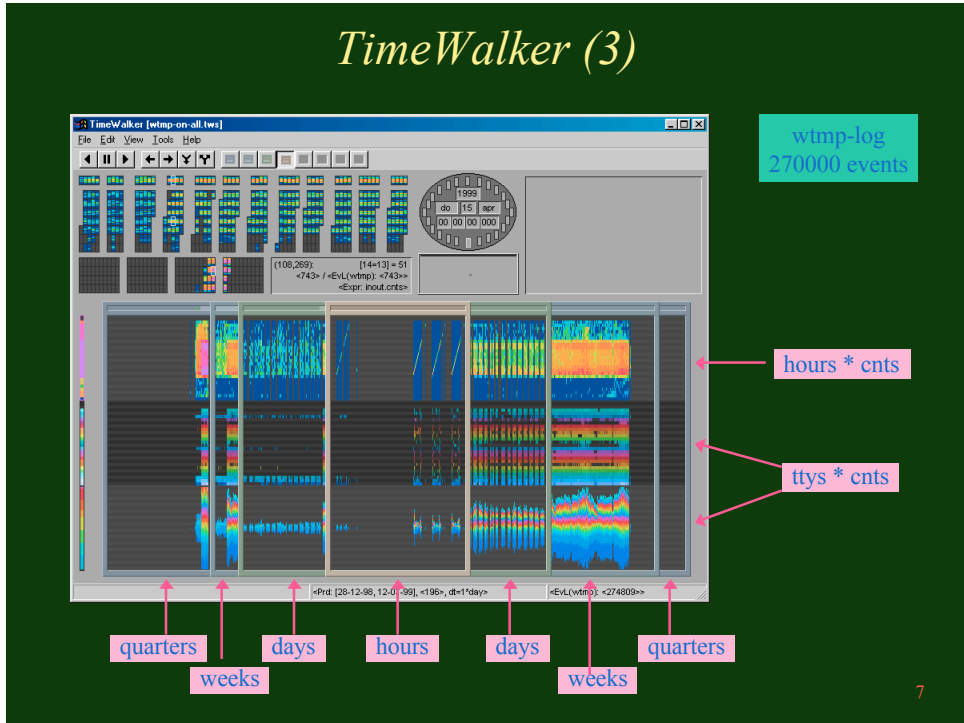
5

TimeWalker (2)

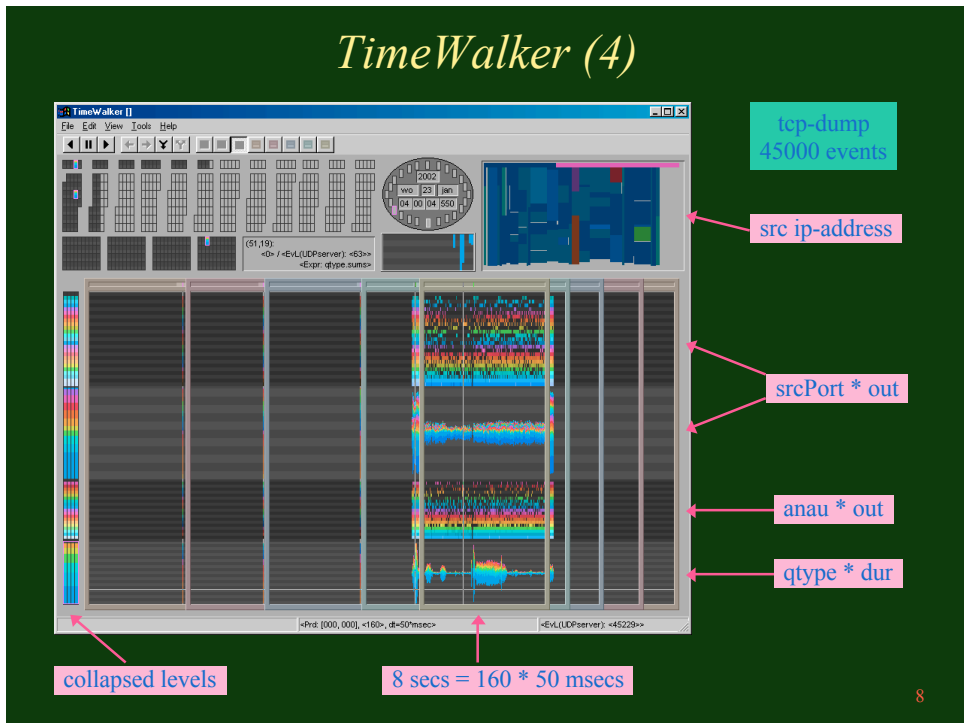
demo

6

TimeWalker (3)



TimeWalker (4)



Tough (?) domain

- data
 - huge amounts of events (= timestamped records)
 - inhomogeneous formats
- graphics
 - maximized information density down to pixel level
 - real-time zooming and animation
- visual data-mining
 - automatic disclosure of the unknown
 - context + detail without scrolling (or hiding)
- generic (y,z) dimensions
 - flexible and simple
 - scalable and fast

9

Wide-spectrum scope

- sketch
- specification
- documentation
- implementation
- persistent data
- testing & refactoring
- deployment
- IDE
- presentation

10

Specification (1)

```
timeModel.py
1
2 + if 1: # imports
9 - class Time (wx.wxDateTime):
10 +     def __init__ (self, v=None, format=None):
12 +     def __getstate__ (self):
14 +     def __setstate__ (self):
16 +     def __float__ (self):
18 +     def __str__ (self):
20
21 - class Period:
22 +     def __init__ (self, t1, t2=None, unit='day', delta=1, size=0):|
36 +     def __len__ (self):
39 +     def __getitem__ (self, i):
47 +     def __getslice__ (self, i, j):
54 +     def __contains__ (self, tp):
63 +     def __cmp__ (self, p):
69 +     def __str__ (self):
72 +     def reset (self, t, size=None):
101 +     def shift (self, n):
104 +     def index (self, t):
129
```

11

Specification (2)

```
metaModel.py
29
30 - class Ref (object):
31 +     def __init__ (self, value):
48 - class Element:
49 +     def __init__ (self, **args):
71 - class Error (Dict):
72 +     def log (arg):
84 +     def __init__ (self, value, tb=1, beep=1, show=1):
102 - class MetaRecord (Dict):
103 +     def __init__ (self, name, fieldSpecs=[]):
143 - class FieldDescr (Record):
144 +     def __init__ (self, fspec):
204 - class Record (List):
205 +     def __init__ (self, metaRecord, record):
244 - class RecordList (List):
245 +     def __init__ (self, metaRecord, records=None,
246 +                 indices=None, shareData=0,
247 +                 sortIndex=None, sorted=0):
248 +     ...
363 - class Expr (object):
364 +     def __init__ (self, expr, metaRecord, context={}):
412
```

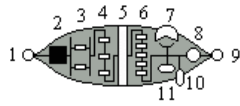
12

Specification (3)

```

+class Tracer:
+class Distributor:
+class Activator:
+class Hypothesis:
+class Assumption (Hypothesis):
+class Expectation (Hypothesis):
+class Pillow:
+class TestBed:
+class TestSession:
+class TestDossier:
+class TestArchive:
+class ProofChamber:
    
```

ProofChamber



1 incoming trigger connector	5 trigger quantity indicator	9 outgoing trigger connector
2 icon targetClass	6 (and,or)-gate postcondition	10 external exception indicator
3 (and,or)-gate incoming hits	7 timeout indicator	11 internal exception indicator
4 (and,or)-gate precondition	8 state summary indicator	

13

Specification (4)

```

+class Thing:
+class Set:
+class Hierarchy:
+class Artifact (Thing):
+class Actor (Thing):
+class Troupe (Actor, Set):
+class Forum (Troupe):
+class Domain (Thing, Hierarchy, Set):
+class Activity (Thing):
+class Goal (Thing, Set):
+class Privilege:
+class Responsibility (Domain, Privilege):
+class Capability (Domain, Privilege):
+class Role (Domain, Privilege):
+class Constraint (Thing):
+class Type (Thing):
+class AntiType (Type):
+class Space (Domain):
+class Gate (Thing):
+class Scope (Thing, Hierarchy):
+class Action:
+class Log:
+class Context:
    
```

RoleKeeper

```

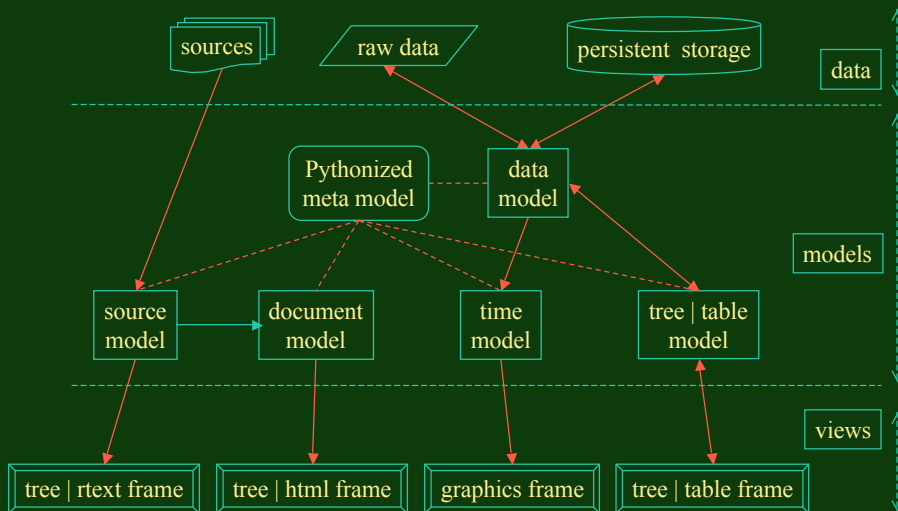
transitions = {
    'look': 'SEEN',
    'use': 'USED',
    'modify': 'MODIFIED',
    'include': 'WIDENED',
    'remove': 'NARROWED',
    'create': 'EXPANDED',
    'validate': 'VALIDATED',
    'invalidate': 'INVALIDATED',
    'approve': 'APPROVED',
    'disapprove': 'DISAPPROVED',
    'audit': 'AUDITED',
    'delegate': 'DELEGATED',
    'revoke': 'REVOKED'
}
    
```

14

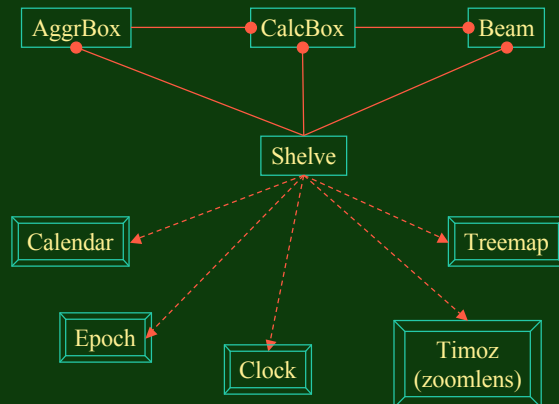
Documentation

demo

Implementation (1)



Implementation (2)



17

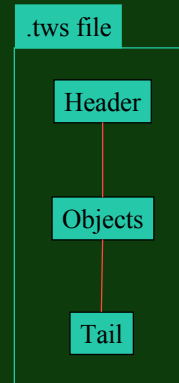
Implementation (3)

#mods	#lines	avg	max
59	15950	270	780

18

Persistent data

- Python pickling mechanism
 - fast & compact (100 bytes / rc)
 - transparent persistency
- Complete snapshots
 - shelve
 - bundle
 - treemap
 - calc boxes
 - aggr boxes
 - data
 - options



19

Testing & refactoring

```
runModule.py
11
12 -if 1: # imports
13     import unittest
14     import profile
15 +
16     ...
30 -class RunModule:
31     def __init__(self, module='__main__',
32                 args=None, mode=None,
33                 verbosity=1):
34 +
35     ...
236 +class TwTestLoader (TestLoader):
275 +class TwTextTestRunner (TextTestRunner):
308 +class PyShellFrame (wx.wxFrame):
340 +class WidgetFrame (wx.wxFrame):
410
411 -if __name__ == '__main__':
412     RunModule(mode='test', verbosity=2)
413
```

20

Deployment

- Distutils
 - bdist => .exe, .rpm
 - relative pathnames ?
 - data, examples ?
 - environment settings ?
- Platform dependencies
 - fonts
 - widget shape & behaviour
 - keyboard & mouse events
 - library versions
 - *hidden* errors

21

IDE

demo

22

Performance

- graphics

```
dc.DrawPointList(pnts, pens)
dc.DrawLineList(lines, pens)
```

- pickling

```
class DataRecordList
    class Col
        class Stretcher
```

- aggregation

- simple types => reduce
- other types => to be redone in C

- builtin methods

- overloading by exception ?

23

Conclusions

- most libraries are legacy-like

- names, types, (user) interfaces
- awful 1-1 mappings from C++
- contra-productive for our dreams

- some name-spaces are not relative

- file-names: Distutils, sharing via mounts
- module-names: unpredictable imports

- postpone coding in C++

- right abstraction => right performance

- honour Python as Zen-master

- not elegant => not (durable, flexible, ...)

24

About Python ...

“Languages that prevent errors are
as feasible as knives that can cut
meat but not hands.”

David Lorge Parnas

25

Colophon TimeWalker

Sponsor: © 2001, Stichting NLnet

License: GNU General Public License

Project: sourceforge.net/projects/timewalker

Author: theo.de.ridder@planet.nl

26