ONERA

CGNS & elsA

Poinot Marc
DSNA/ELSA

poinot@onera.fr
http://elsa.onera.fr

Motivations

CFD standard for data representation is required

We want to use a common data representation for interoperability

We want to ensure our data a long life format

ONERA is a CFD expert

We are one of the aerospace main actors

We want to participate to a standard

- Add our expertise
- Be aware of what's going on

Some customers are asking for CGNS

Common data representation at specification level

Interoperability for application framework

ONERA

CGNS related developments

elsA

Main I/O data representation

- In-memory tree
  - Interoperability / Multi-processing
- Disk tree
  - Validation resources

pyCGNS

A Python binding to CGNS

Gluing language for application framework

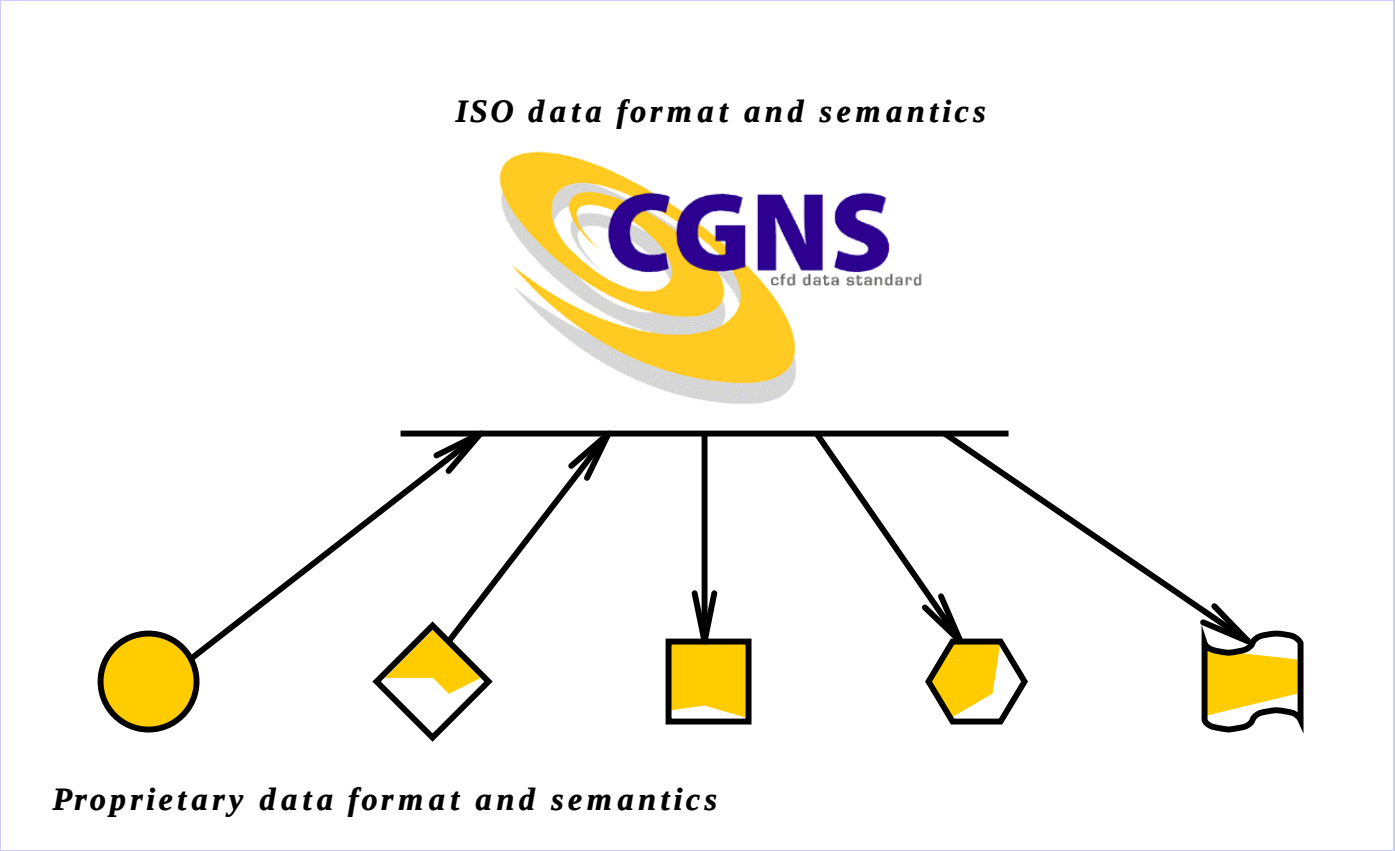
- e.g. translators to CGNS trees
  - From/To Tecplot and Aerospace actors data format

Open source

`http://elsa.onera.fr/CGNS/releases`

ONERA

An ISO data representation



ONERA

Application side features/ concerns - 1

Top level side

Share application data

- Semantics
- Common names/ structure

Self-described tree structure

ISO process

Lower level side

Share physical data

Actual interoperability between applications

CAUTION: NO STANDARDIZATION PROCESS

ONERA

Application side features/ concerns - 2

Missing features

Protocol

- No data lifecycle
- No data ownership

Specific data/ extensions

- Guidelines for proprietary sub-tree

Parallel

- Application side: uses in-memory representation ?

Share/ Transactionnal/ Query

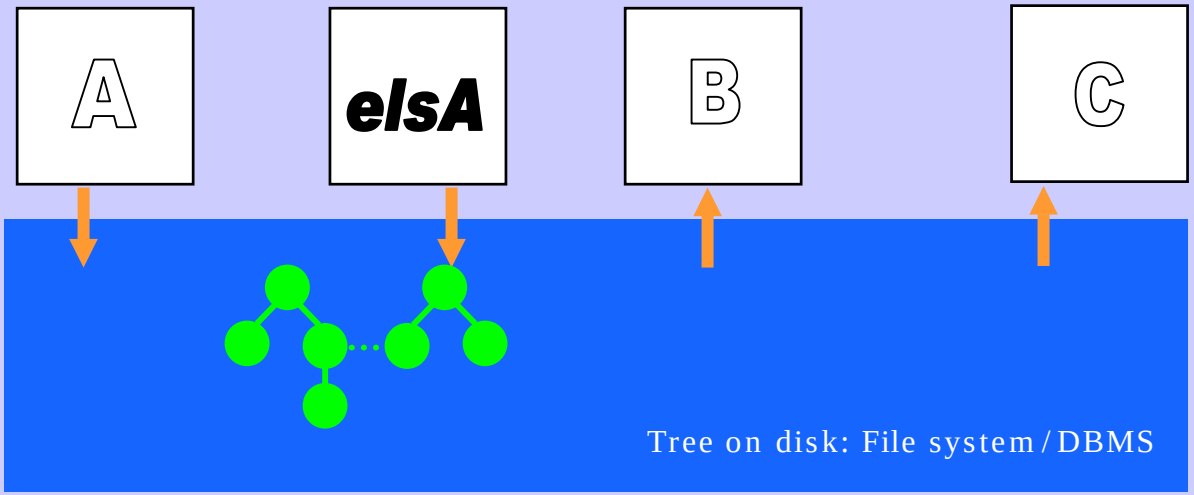
- DBMS uses blobs at application level ?

Other

- Force an XML mapping

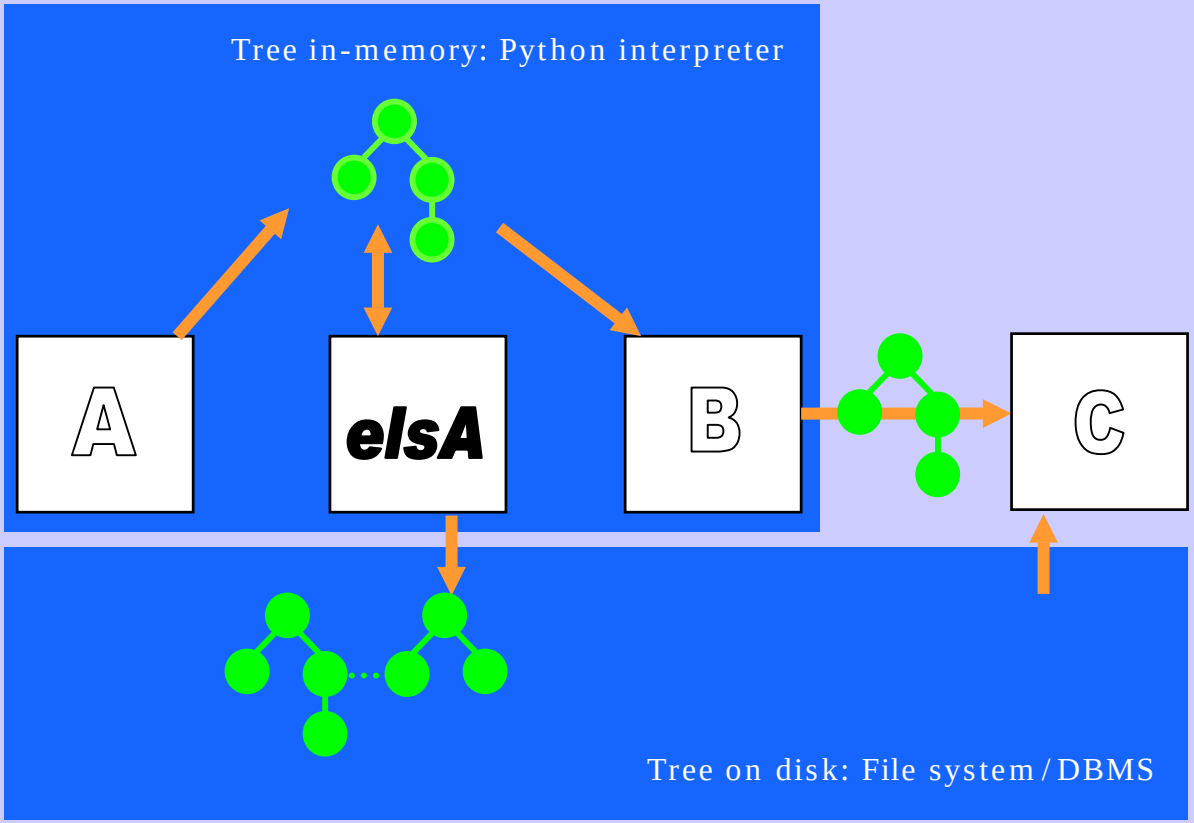
ONERA

Architecture - 1



ONERA

Architecture - 2



ONERA

pyCGNS Status - 1

v0.3

Binding

- Full ADF wrapper
- 60% MLL wrapper
- 10% OOL layer

Python + Numerical Python

- Standard package
- Documentation+ test suite

Tools

- *SIFT* - SIDS instance description check
- *ParseTree* - Tree dump/display as XML tree
- *StampNode* - Add sub-nodes with specific stamps (date, ownership…)

http://elsa.onera.fr/CGNS/releases

ONERA

pyCGNS Status - 2

v0.4 *(1Q2002)*

In-memory representation

- Python/C/C++/Fortran in-memory tree sharing

Tools

- SIFT - Generate data tree from SIDS instance description

v1.0 *(3Q2002)*

Binding

- Full MLL wrapper
- Full OOL layer

Documentation

- Features 100% documented + tutorial
- Design partly documented

Test suite

- Complete test suite for OOL/MLL/ADF

ONERA

Code example - pyCGNS

```
#!/usr/bin/env python
# CFD General Notation System - CGNS lib wrapper
# ONERA/DSNA/ELSA - poinot@onera.fr
# User's Guide to CGNS - C.L.Rumsey et al. examples translation
#
from CGNS      import midlevel
from CGNS.wrap import *
from Numeric   import *
#
#   open CGNS file for read-only
file=pyCGNS('grid.cgns',midlevel.MODE_READ)
index_base=1
index_zone=1
index_flow=1
isize=file.zoneread(index_base,index_zone)[3]
irmin=[1,1,1]
irmax=[isize[0],isize[1],isize[2]]
(loc,name)=file.solinfo(index_base,index_zone,index_flow)
if (loc != midlevel.Vertex):
  print "Error, GridLocation must be Vertex!  Currently:",
  print midlevel.GridLocation[loc]
#   read flow solution
r=file.fieldread(index_base,index_zone,index_flow,'Density',midlevel.RealSingle,irmin,irmax)
p=file.fieldread(index_base,index_zone,index_flow,'Pressure',midlevel.RealSingle,irmin,irmax)
#   close CGNS file
del file
print "Successfully read flow solution from file 'grid.cgns'"
print "For example, r,p(21,17,9)=",r[20,16,8],p[20,16,8]
print "Program successful... ending now"
```

ONERA