SEVEN KEYS FOR PRACTICAL UNDERSTANDING AND USE OF CGNS

Marc Poinot, SAFRAN
Christopher Rumsey, NASA LaRC
AIAA SciTech 2018
Motivation

CGNS is well known in the CFD community, but:
- often mismatched with its mid-level library
- only understood as mesh storage on disk

We have seen some initiatives to develop a new standard to:
- define an implementation-independent data model
- add Computer-Aided Design (CAD) related features
- implement in HDF5

➤ This looks quite close to CGNS

We propose to gather initiatives rather than creating entropy: let us refresh our comprehension…

Common comments:
- CGNS is a library
- CGNS cannot fit my own data structure
- There are too many ways to describe the same feature in CGNS
- CGNS cannot handle parallel processing
- CGNS is only for archival data
- CGNS files are too big
- The recommended implementation is HDF5 but only ADF is available
- There are few useful tools for CGNS manipulation
- I already have HDF5 and I do not need CGNS
- CGNS inefficiently stores time data and statistics
### CGNS entities and their acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Meaning</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGNS</td>
<td>CFD General Notation System</td>
<td>Identifier</td>
</tr>
<tr>
<td>CGNS/SC</td>
<td>Steering Committee</td>
<td>Group of people</td>
</tr>
<tr>
<td>CGNS/SIDS</td>
<td>Standard Interface Data Structure</td>
<td>Textual document</td>
</tr>
<tr>
<td>CGNS/CPEX</td>
<td>CGNS Proposal for EXtension</td>
<td>Procedure and textual document</td>
</tr>
<tr>
<td>CGNS/FMM</td>
<td>File Mapping Manual</td>
<td>Textual document</td>
</tr>
<tr>
<td>CGNS/HDF5</td>
<td>HDF5 implementation</td>
<td>Textual document</td>
</tr>
<tr>
<td>CGNS/Python</td>
<td>Python implementation</td>
<td>Textual document</td>
</tr>
<tr>
<td>CGNS/MLL</td>
<td>Mid-Level Library</td>
<td>Software library</td>
</tr>
</tbody>
</table>
Seven keys

1. Abstract model
   - What is the data I want to define?
   - How do I define it?

2. Interoperability
   - How to go from CAD to mesh, to post-processing, visualisation and then back to CAD?
   - How to exchange data during CFD/CSM, CFD/CAA or any multiphysics workflows?

3. Functional coverage
   - Do I have enough functions to describe my data?
   - What is the remaining volume data I need to define/store?

4. High performance
   - Is there a High Performance Computing (HPC) efficient representation for my data model?
   - Who is going to support it?

5. Versatile implementation
   - Can I manipulate my data in a flexible way?
   - Do I have to re-invent all the functions I need for my data processing?

6. Extensibility
   - How can I describe out-of-CFD data, such as chemistry, high order, CSM features, new CFD trends?
   - Does the standard belong to NASA?

7. Open System
   - Can I connect the simulation elements together as black boxes?
   - Is there a mandatory implementation?
A model is what you want to specify but not how you want to implement it.

Informal definition, textual non-exhaustive description:

Formal definition, supported using a grammar:

**SIDS type definitions**

**Types and Indexing conventions**

Abstract model

**5.1 Definition: DataArray_t**

DataArray_t describes a multi-dimensional data array of given type, dimensionality and size in each dimension. The data may be dimensional, non-dimensional or pure constants. Qualifiers are provided to describe dimensional units or normalization information associated with the data.

```
DataArray_t< DataTypes, int Dimension, int [Dimension] DimensionValues > : = 
{ 
  List( Descriptor_t Descriptor1 ... DescriptorN ) ; 
  Data( DataTypes, Dimension, DimensionValues ) ;
  DataClass_t DataClass ;
  DimensionalUnits_t DimensionalUnits ;
  DimensionalExponents_t DimensionalExponents ;
  DataConversion_t DataConversion ;
} ;
```

**Notes**

1. Default names for the Descriptor_t list are as shown; users may choose other legitimate names. Legitimate names must be unique within a given instance of DataArray_t and shall not include the names DataClass, DimensionalUnits, DimensionalExponents, or DataConversion.
2. Data() is the only required field for DataArray_t.
Interoperability

Families are the means to hold user’s information

- Topology entities, CAD reference
- Same entity for different models
- Entities with special features
- Simulation specialized entities

Simultaneous families allowed
Hierarchy of families

- Insure traceability through processes
- Allows high level interoperability

The highest level of interoperability is the user’s meaning
The lowest is the implementation
Functional coverage

The CGNS tree changes all along the simulation workflow

- It covers our CFD needs and more
- It doesn't cover everything, but it could eventually fulfill any needs: CPEX

99% of the volume of the data already is defined
High performance

HPC implementation

- HDF5
  - HDF Group support for portage/tuning
  - Very large data but limited number of types
  - Fast memory/disk swap
  - Smart memory mapping
  - Parallel features

- The HDF5 implementation specification is public
  - Write your own application
  - Specific node orientation
  - Specific memory mapping (interlaced, non-contiguous, partial, out-of-core…)

AIAA-SciTech 2018 / Seven keys for practical understanding and use of CGNS
Versatile implementation

Gluing/prototyping the easy way

- Python/numpy
  - Your C/Fortran array IS your python array
  - Reuse of the very large python libraries
  - No-class paradigm, easy serialization/deserialization
- Memory to memory exchange (same process, inter-process, inter-nodes, inter-hosts...)
- Fast creation/modification of CGNS trees

```python
import CGNS.MAP as CGM
import CGNS.PAT.cgnsutil as CGU
import CGNS.PAT.cgnskeywords as CGX
import CGNS.PAT.cgnslib as COL

(T,L,P)=CGM.load('HL-CGM.cgns')

familiesSet()
pathlist=CGU.getAllNodesByTypeOrNameList([CGX.CGNSTree_ts,'Base',CGX.Zone_ts,CGX.ZoneBC_e,CGX.BC_ts])

for path in pathlist:
    botype=node[1]
    family=path[2]

    for name in families:
        node=CGU.getNodeByPath(T,'Base')

    base=CGU.getNodeByPath(T,'/Base')
    for family in families:
        base=CGU.getNodeByPath(T,'Base')

    COL.newFamily(name,family)
    COL.newBC(fam,'BC'+family)

CGM.save('HL-CGM.cgns',T)
```

Change all BCs to family defined
Extensibility

- **A support for user needs**
  - **CPEX**
    - Propose your own extension
    - Do not break already used features
  - **Examples**
    - NGON HPC redefinition
    - Arbitrary High order
    - Enhanced Chemical data model
    - Arbitrary Reference Frame
  - **PROS**
    - Arguing often leads to better ideas
    - Once adopted, everyone uses your data model / implementation
  - **CONS**
    - Lengthy process
    - You have to explain your ideas

AIAA-SciTech 2018 / Seven keys for practical understanding and use of CGNS
Open system

Specification & connection only through public interfaces

- High Level: CGNS/SIDS
- Low Level: CGNS/FMM
- Implementation: CGNS/HDF5, CGNS/Python

- Any compliant implementation can 'connect' to others
- Dedicated implementation is required for HPC

Diagram:

- CGNS/SIDS
  - data model
- CGNS/FMM
  - per-node mapping
- CGNS/MLL
  - CHLone
- CGNS/ADF
- CGNS/HDF5
  - single-node mappings
- CGNS/Python
  - your own implementations
  - pyCGNS

AIAA-SciTech 2018 / Seven keys for practical understanding and use of CGNS
## Answers to remarks

| ☑️ | CGNS is a library |
| ☑️ | CGNS cannot fit my own data structure |
| ☑️ | XML is the way to describe data |
| ☑️ | **Extension process is too long for me** |
| ☑️ | There are too many ways to describe the same feature in CGNS |
| ☑️ | CGNS cannot handle parallel processing |
| ☑️ | HPC cannot waste time on data model |
| ☑️ | CGNS is only for archival data |
| ☑️ | **CGNS files are too big** |
| ☑️ | The recommended implementation is HDF5 but only ADF is available |
| ☑️ | There are few useful tools for CGNS manipulation |
| ☑️ | I already have HDF5 and I do not need CGNS |
| ☑️ | **CGNS inefficiently stores time data and statistics** |

| ☐️ | CGNS is only for CFD |
Conclusion

You now have a broader view of CGNS

CGNS is far more than a library ➔ it is a means for CFD workflow interoperability

Do you want to add CAD traceability?
Do you have such a standard for CSM, FSI, CAA, or other field?
Do you want more data structure, more dedicated implementations?

Join www.cgns.org
cgnstalk@lists.nasa.gov