HSF Meeting December 4 2019

Overview of Detector Description in ATLAS

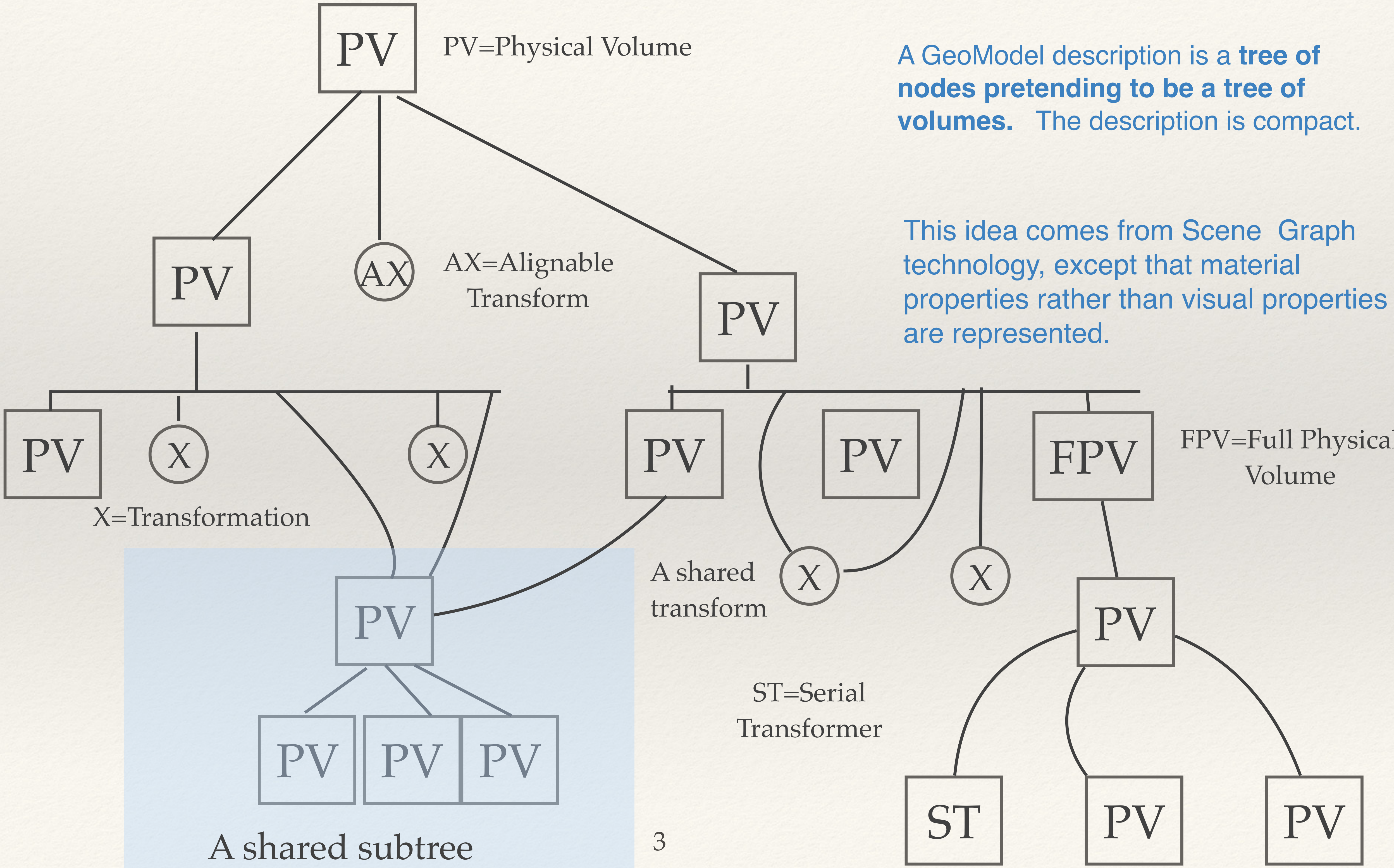
Joe Boudreau for the ATLAS Experiment

Contact: geomodel-developers@cern.ch

Components:

- The GeoModel Toolkit, *in use in ATLAS for 16 years, one of the most stable components of the software stack.*
- The Detector Description Database, *for primary numbers, held in Oracle and replicated to remote sites in an SQLite file.*
- Detector Factories & databases used by subsystems, *which construct a GeoModel software description of the detector from subsystem specific input sources.*
- Readout Geometry, *derived from the GeoModel description.*
- Visualization tools *to explore geometries in great detail*.
- New components under design by a re-activated Detector Description group which will be described in the second half of this talk.

The GeoModel Toolkit



A GeoModel description is a **tree of nodes pretending to be a tree of volumes.** The description is compact.

This idea comes from Scene Graph technology, except that material properties rather than visual properties are represented.

Memory footprint and initialization time

System	Memory (Mb)	CPU (s)
Pixel	12.8	0.35
SCT	13.3	0.14
TRT	22.2	0.18
LAr	32.3	1.45
Tile	14.7	0.36
Muon	36.1	0.79
Total	**131.4**	**3.27**

In more detail

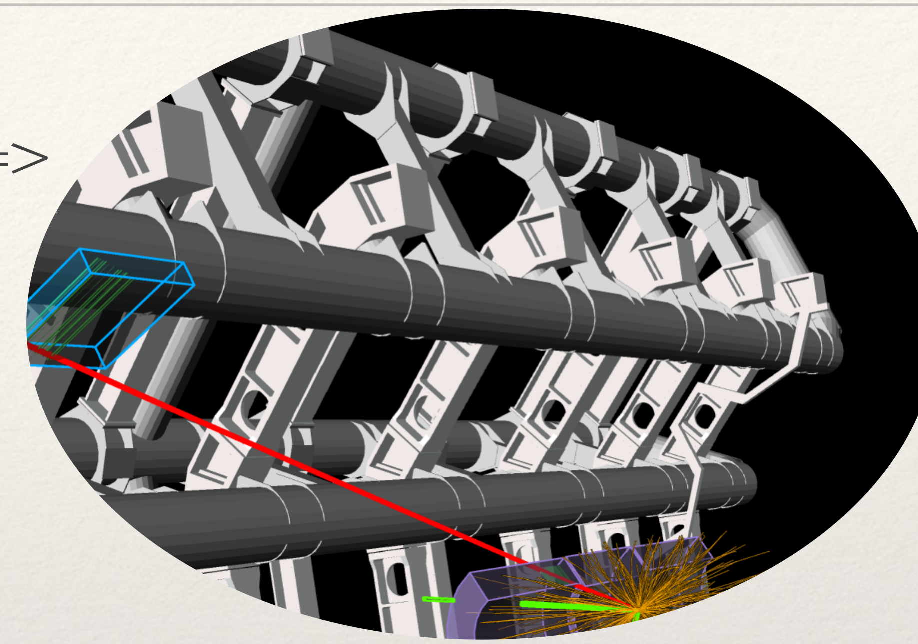
Logical volumes

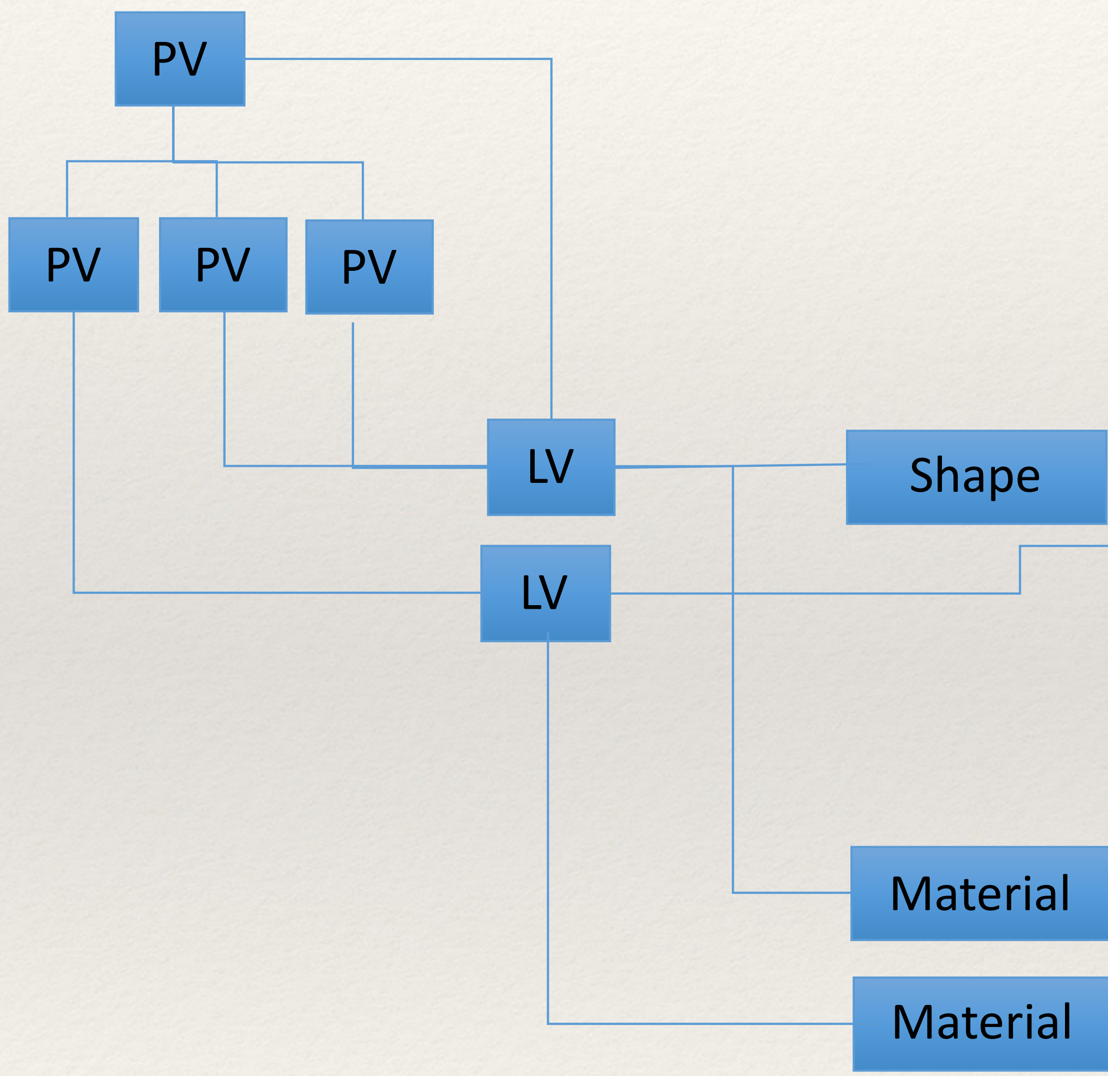
Shapes

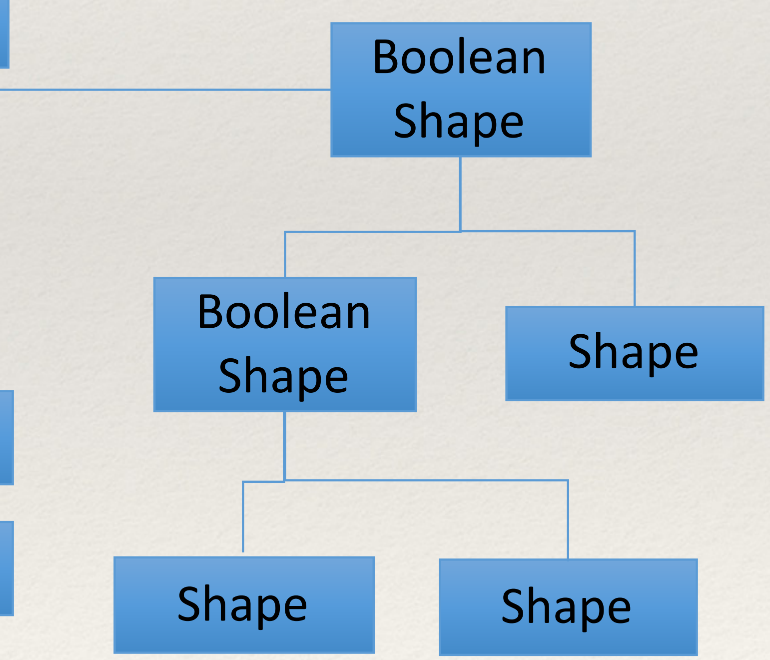
Materials

Boolean Shapes

Examples of
Boolean shapes==>







All instances of physical volumes, transformations, logical volumes, shapes, materials, Boolean shapes.. may be shared. Memory management through reference counting (this vestige of the olden days to be removed…).

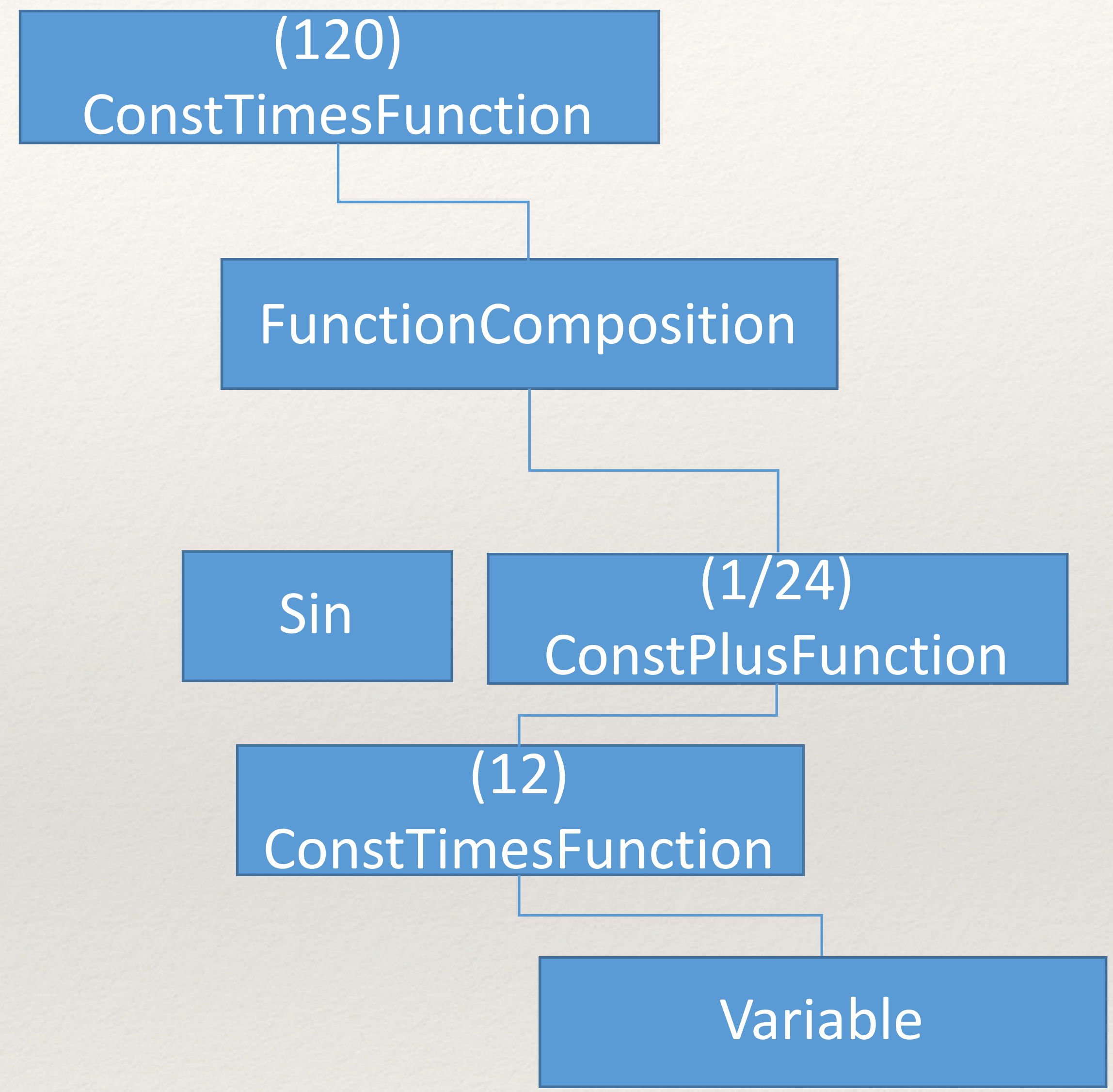
The GeoModel Toolkit:

- Depends on Eigen
- Is merely a geometry layer; it does not *do* anything.
- Allows for save & restore of a geometry description (currently in SQLite format*, file size is 43MB)
- Has extensive memory optimizations:
  - shared instancing
  - serial transformers
- However, I am not sure that all of the subsystems experts have availed themselves aggressively of the optimization techniques.

*this introduces a dependency on QtSQL, a solution adopted to speed up the development cycle. That dependency is temporary and will soon disappear.

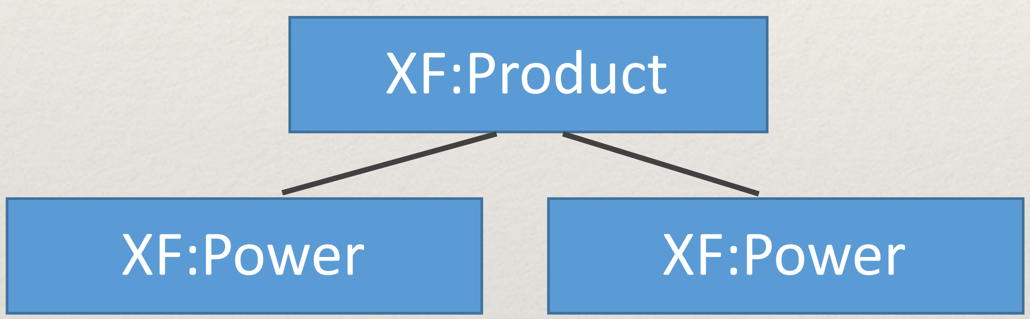
Serial transformers:

$$120 * \sin\left(12 * i + 1/24\right)$$



In the geometry kernel: A totally generic recipe for parameterizing transformations (of physicial volumes, including subtrees):

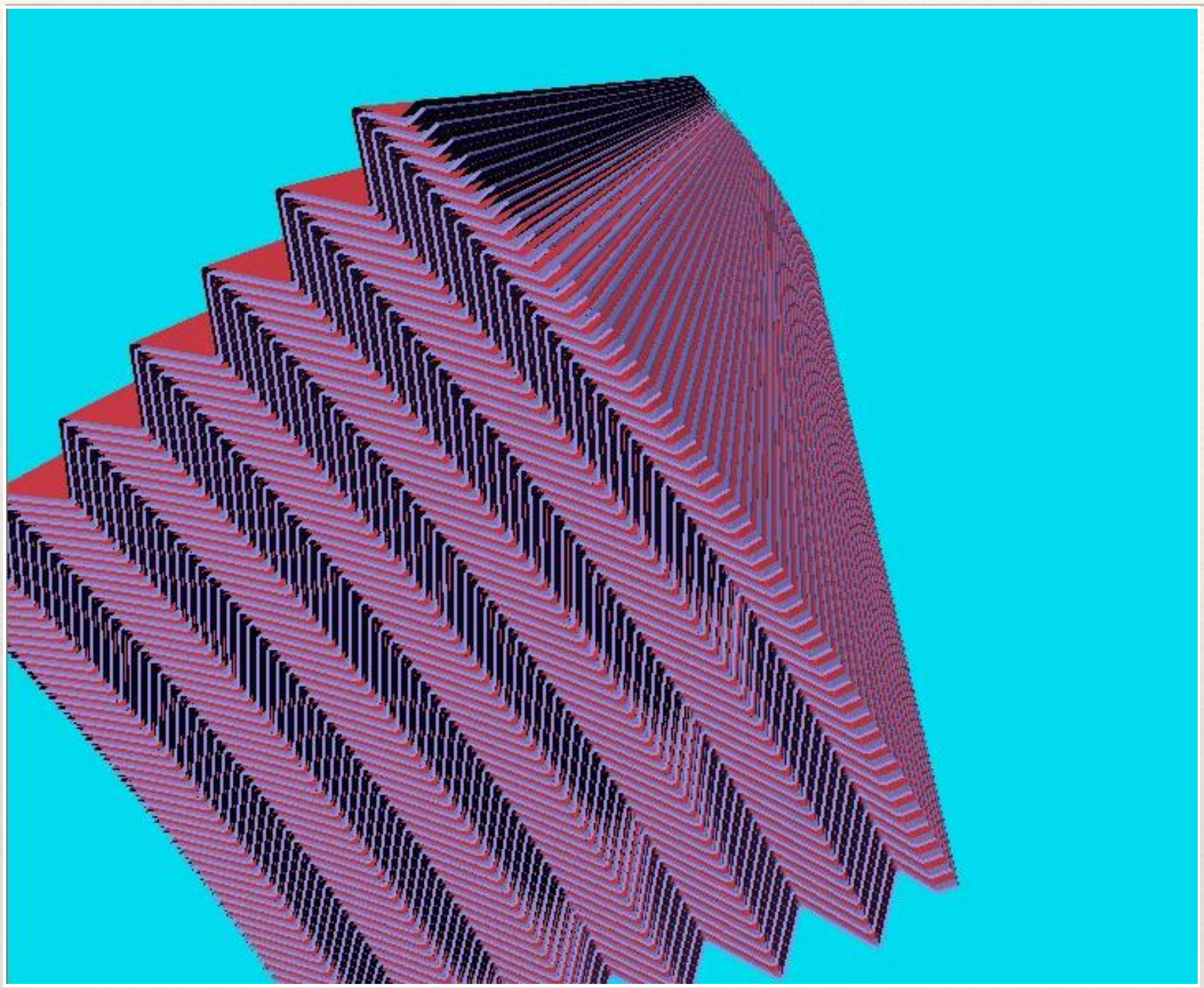
$ROT_x^{f(i)}*ROT_y^{g(i)}*TRANS_x^{h(i)}$..... & cetera

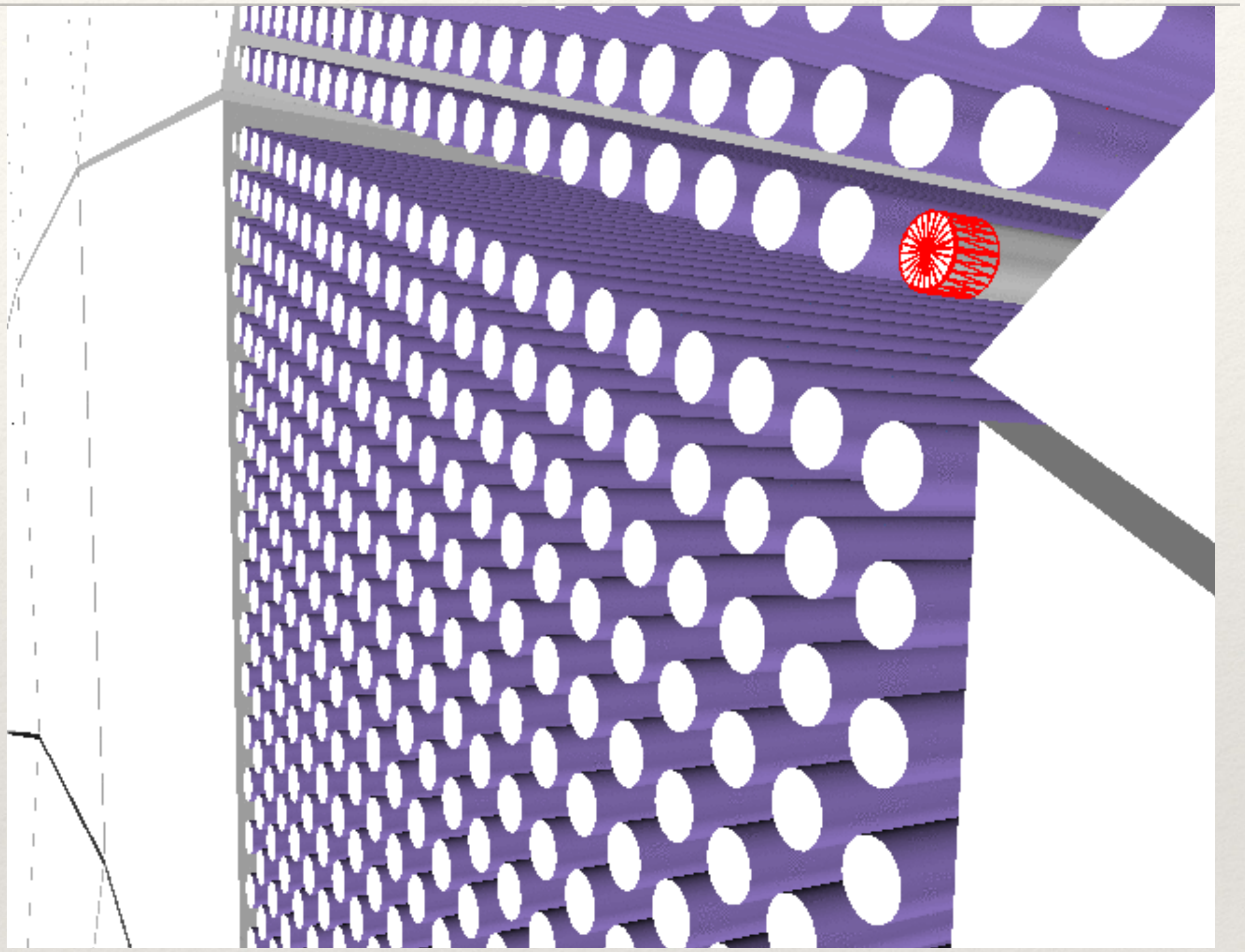



```
const unsigned int NPLATES=100;
Variable      i;
Sin           sin;
GENFUNCTION f = 360*deg/NPLATES*i;
GENFUNCTION g = -sin(4*f);
TRANSFUNCTION t1 = Pow(RotateZ3D(1.0), f)*
                   TranslateX3D(1100*cm)*
                   Pow(TranslateZ3D(800*cm),g);
```

These mathematical expressions are saved along with the rest of the geometry when it is persistified to SQLite.

Many of the challenging geometries are implemented as SerialTransformers…





Interestingly, we have also implemented a G4Parameterization using the same trick.

Now we can convert during Geo—>G4 conversion *at the flick of a switch*:

- Parameterization to placements, or
- Parameterization to parameterization.

Past experience has shown that the former choice is favorable to simulation, but will this hold true on the "new architectures".. ?

Re-activation of the detectors description group which was retired in 2005.

- Goals:
  - Streamline the detector description workflow.
  - Harmonize the subsystem-specific technologies (mostly XML-based).
  - Address new detector subsystems for Run 4
  - Critically re-examine existing detector description code for possible improvement.
  - Restore expertise in subsystems where it appears to be lacking.

There have been two major impediments to efficient software description:

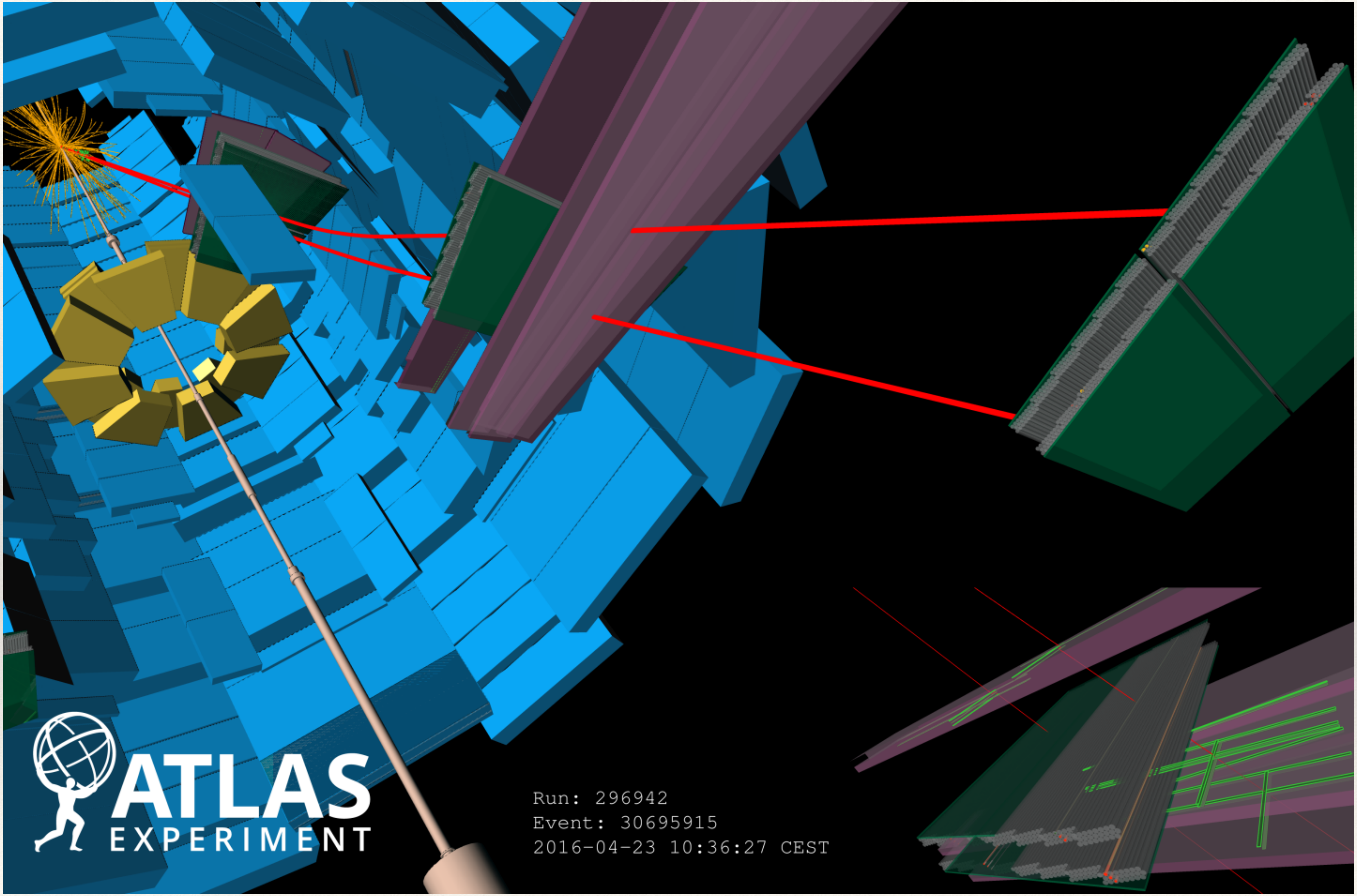
- The database of primary numbers.
- The (Athena*-based) visualization.

Both of these were fine solutions for their time, which was 16 years ago when Athena was not yet burdened with massive functionality, now we can do better….

*Athena is the ATLAS analysis framework, based on GAUDI.

The database

- Problems:
  - The current solution for the storage of primary numbers is a relational data base whose master copy lives in Oracle. The database permits tagging of geometries. But, few people are authorized to write to the database and prototyping is difficult. The access to this database is through Athena.
  - In addition to the database, there are 3 XML parsers in use, some (all?) reading blobs from Oracle or from its replica.
- Solutions being pursued:
  - Transfer all the data to XML files that live in a git repository.
  - Use the git functionality to tag the geometry description.
  - Attempt to unify subsystems under one XML technology.


ATLAS
EXPERIMENT
Run: 296942
Event: 30695915
2016-04-23 10:36:27 CEST

The visualization

- "Virtual Point 1" or VP1 is the authoritative event display and geometry display in ATLAS: it sees the same geometry that goes to GEANT, and the same physics objects that reconstruction & analysis sees….
- …because it runs within Athena. This is both a blessing and a curse.
- The curse:
  - Long initialization
  - Super-heavy dependence on an enormous software stack.
  - Lack of portability
  - Fragility.

Streamline the workflow:

- We make it simpler to change numbers in the database by switching to human-readable XML files rather than Oracle. (an anticipated development)
- We derive, from VP1, a new product called the Geometry Explorer (gmex) with fast failsafe initialization and fully extract it from Athena.
    - this is possible thanks to the recent development of geometry persistification
        - Save & Restore the full geometry description (numbers, parameters, materials, nodes, and relations).
    - reads geometry dump files (current)
    - loads detector factories and displays them (anticipated)
- This is a WYSIWYG geometry editor and quite near to a CAD program for detector geometry.
- This is now part of a "GeoModel tool suite" that will be completely external to ATLAS and licensed to the public. See https://gitlab.cern.ch/GeoModelDev

New "Geometry Explorer": install and run

- Ubuntu instructions (for bionic, disco, and eoan releases):

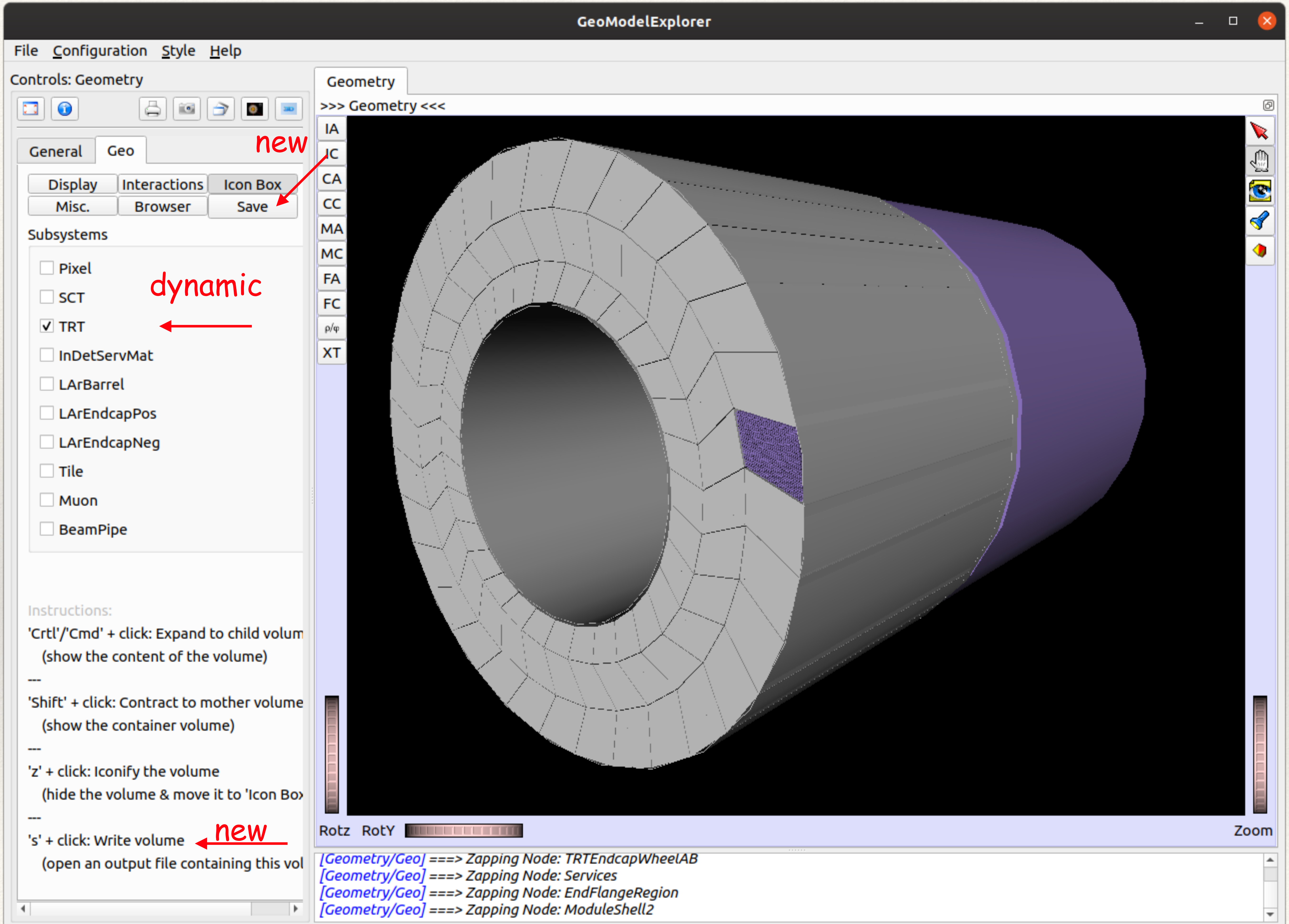
```
• sudo add-apt-repository ppa:kaktusjoe/geomodel
• sudo apt-get update
• sudo apt install geomodel-explorer
• gmex [-d inputFile]
```

- Mac instructions

```
• brew tap atlas/geomodel https://gitlab.cern.ch/GeoModelDev/packaging/homebrew-geomodel.git
• brew install geomodelexplorer
• export GXPLUGINPATH=/usr/local/lib/gxplugins
• gmex [-d inputFile]
```

- For now you can obtain input files from this place:

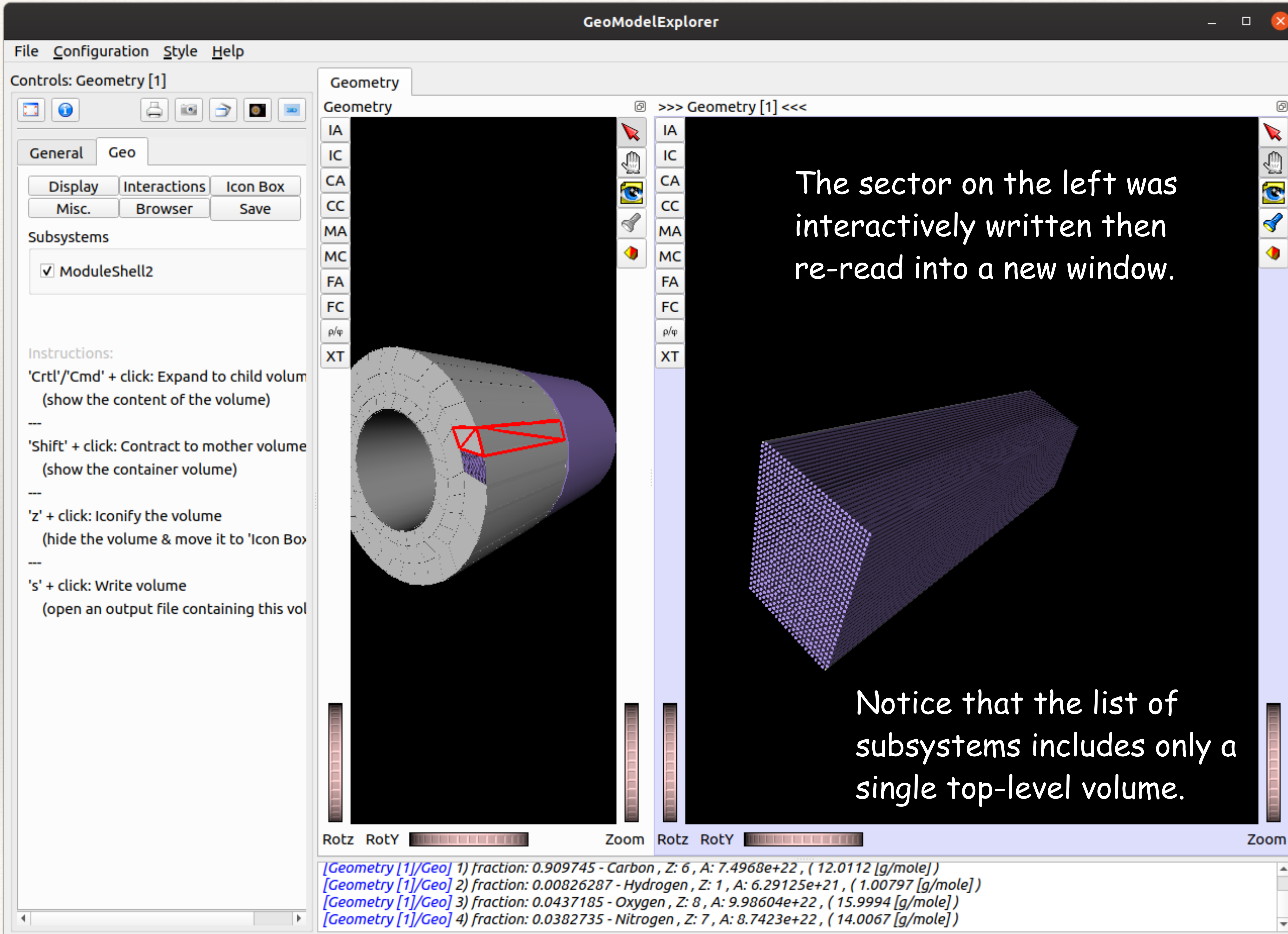
`https://gitlab.cern.ch/GeoModelATLAS/geometry-data/tree/master/geometry`


GeoModelExplorer
File Configuration Style Help
Controls: Geometry
General Geo
Display Interactions Icon Box
Misc. Browser Save
new
Subsystems
Pixel
SCT
TRT
InDetServMat
LArBarrel
LArEndcapPos
LArEndcapNeg
Tile
Muon
BeamPipe
dynamic
Instructions:
'Crtl'/'Cmd' + click: Expand to child volum
(show the content of the volume)

'Shift' + click: Contract to mother volume
(show the container volume)

'z' + click: Iconify the volume
(hide the volume & move it to 'Icon Box

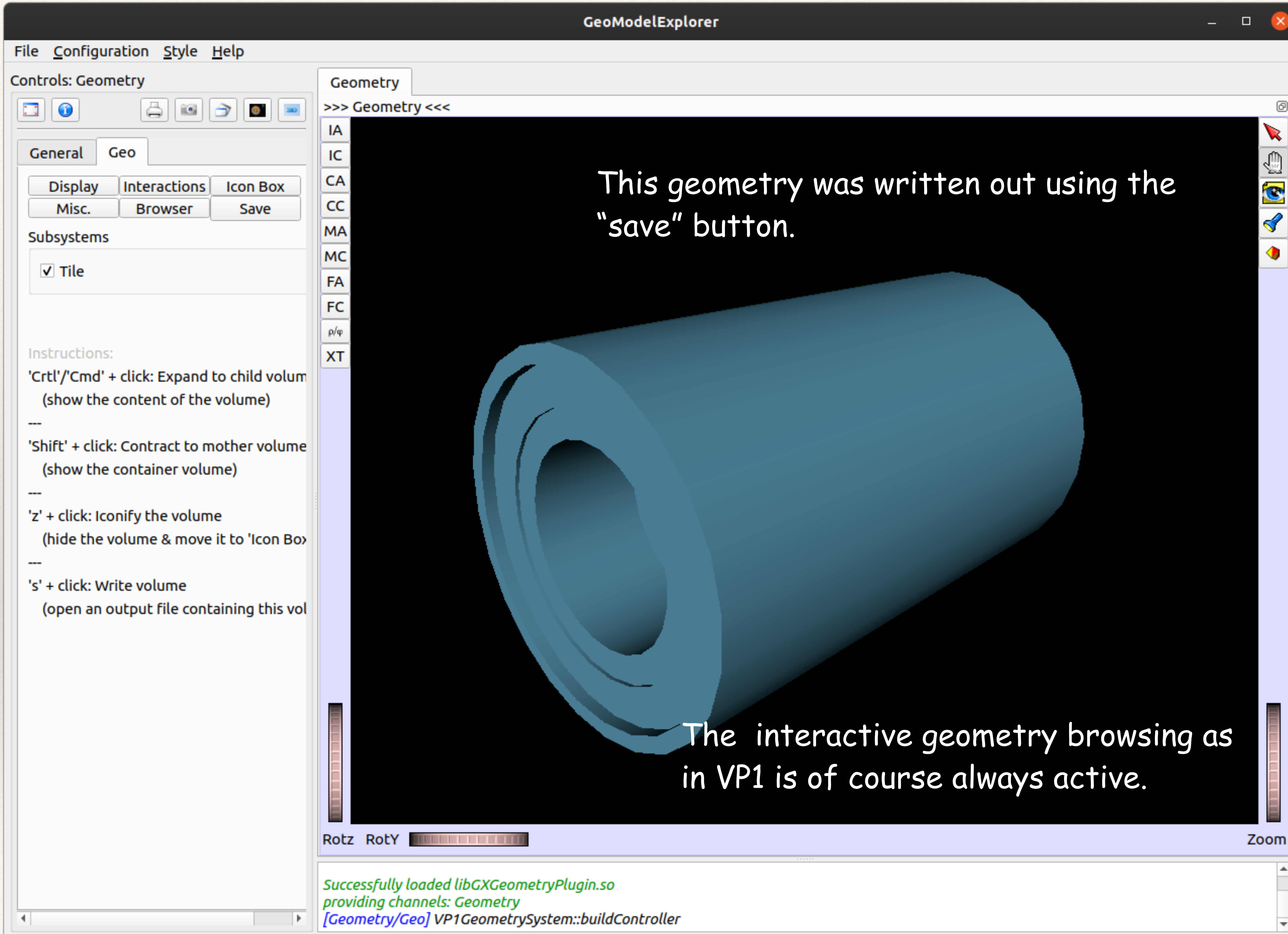
's' + click: Write volume
new
(open an output file containing this vol
Geometry
>>> Geometry <<<
IA
IC
CA
CC
MA
MC
FA
FC
ρ/φ
XT
Rotz RotY
Zoom
[Geometry/Geo] ===> Zapping Node: TRTEndcapWheelAB
[Geometry/Geo] ===> Zapping Node: Services
[Geometry/Geo] ===> Zapping Node: EndFlangeRegion
[Geometry/Geo] ===> Zapping Node: ModuleShell2


GeoModelExplorer
File Configuration Style Help
Controls: Geometry [1]
General Geo
Display Interactions Icon Box
Misc. Browser Save
Subsystems
ModuleShell2
Instructions:
'Crtl'/'Cmd' + click: Expand to child volum
(show the content of the volume)

'Shift' + click: Contract to mother volume
(show the container volume)

'z' + click: Iconify the volume
(hide the volume & move it to 'Icon Box

's' + click: Write volume
(open an output file containing this vol
Geometry
>>> Geometry [1] <<<
IA IC CA CC MA MC FA FC XT
Rotz RotY Zoom
The sector on the left was interactively written then re-read into a new window.
Notice that the list of subsystems includes only a single top-level volume.
[Geometry [1]/Geo] 1) fraction: 0.909745 - Carbon , Z: 6 , A: 7.4968e+22 , (12.0112 [g/mole])
[Geometry [1]/Geo] 2) fraction: 0.00826287 - Hydrogen , Z: 1 , A: 6.29125e+21 , (1.00797 [g/mole])
[Geometry [1]/Geo] 3) fraction: 0.0437185 - Oxygen , Z: 8 , A: 9.98604e+22 , (15.9994 [g/mole])
[Geometry [1]/Geo] 4) fraction: 0.0382735 - Nitrogen , Z: 7 , A: 8.7423e+22 , (14.0067 [g/mole])

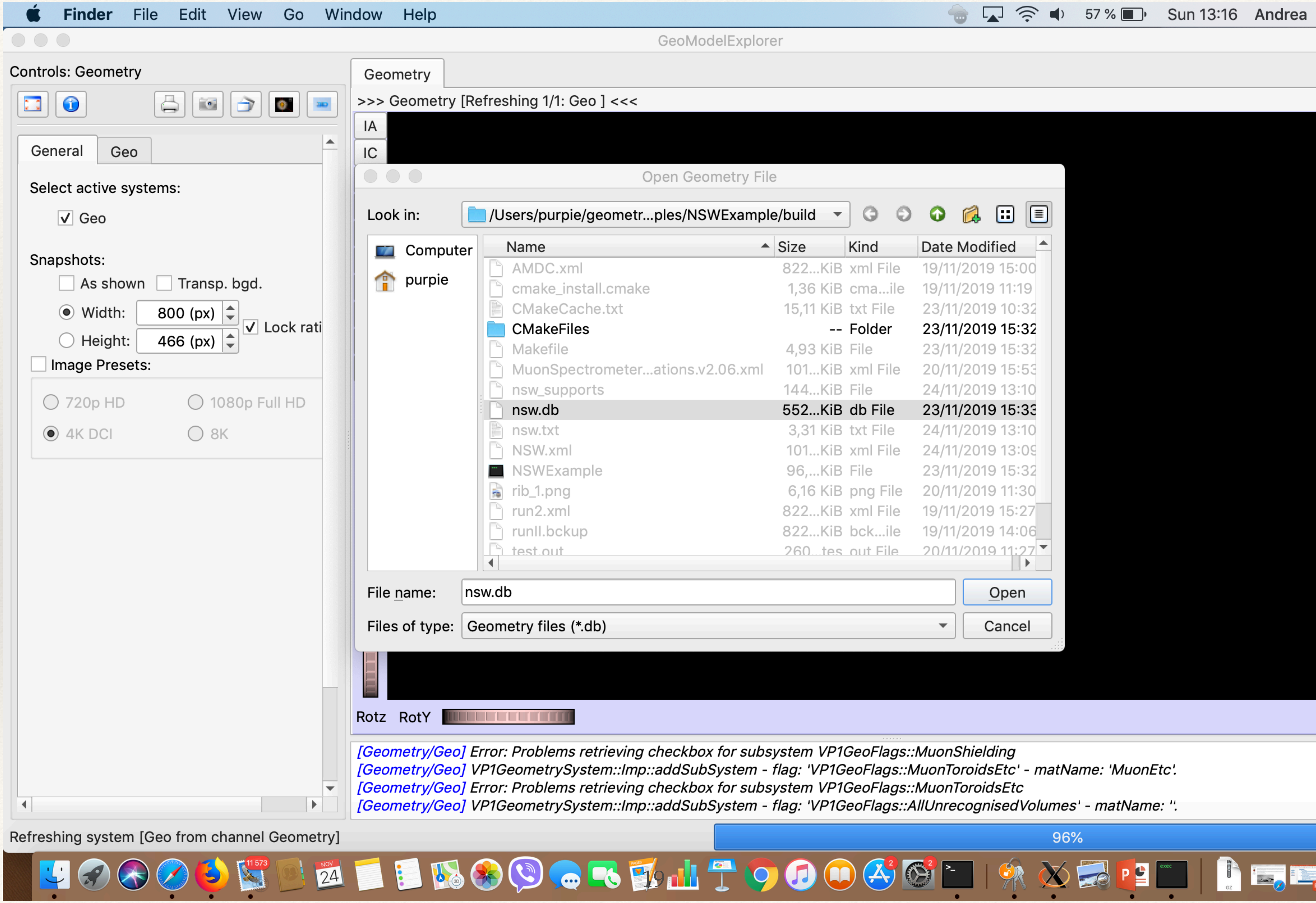

GeoModelExplorer
File Configuration Style Help
Controls: Geometry
General Geo
Display Interactions Icon Box
Misc. Browser Save
Subsystems
Tile
Instructions:
'Crtl'/'Cmd' + click: Expand to child volum
(show the content of the volume)

'Shift' + click: Contract to mother volume
(show the container volume)

'z' + click: Iconify the volume
(hide the volume & move it to 'Icon Box

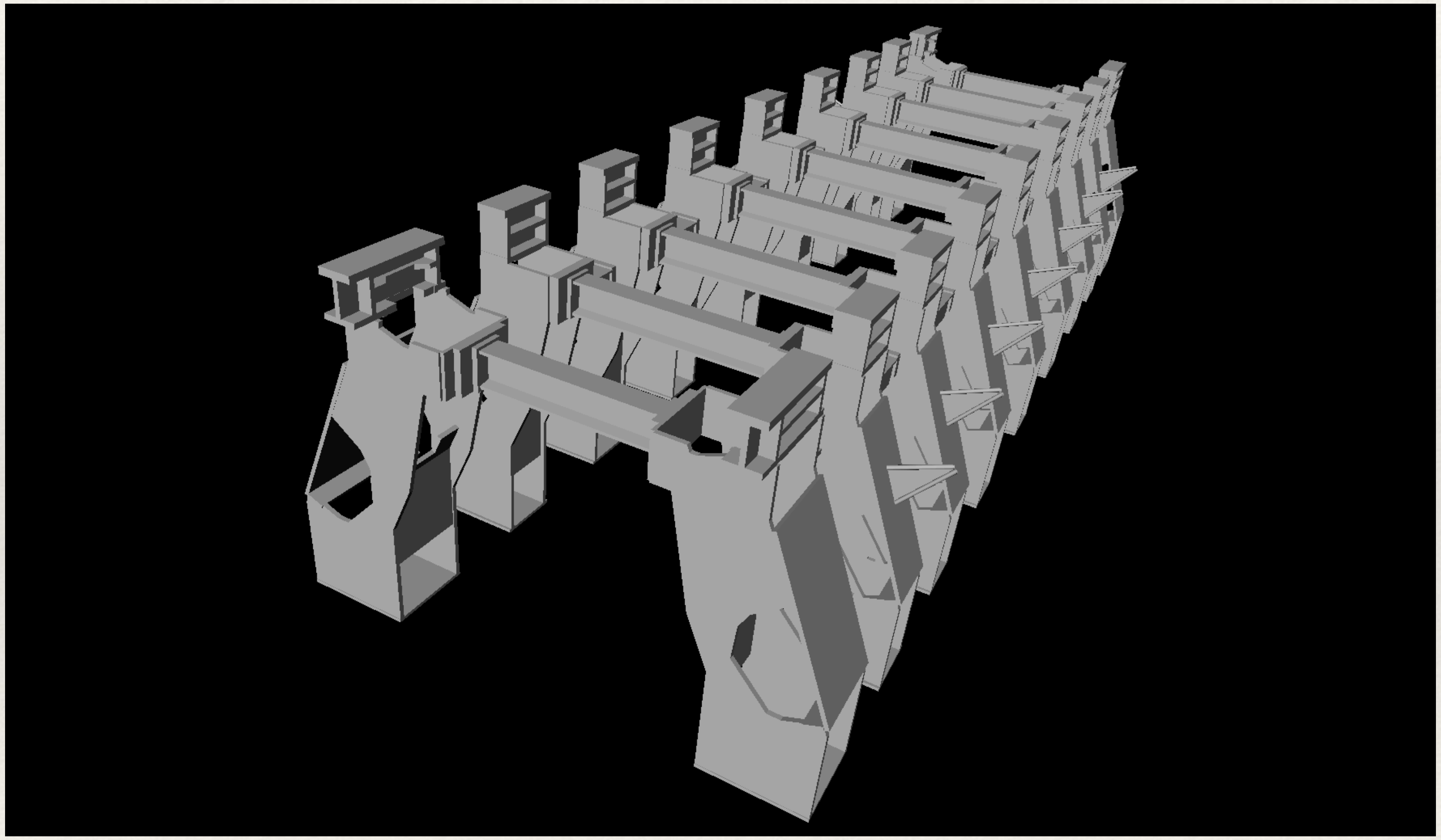
's' + click: Write volume
(open an output file containing this vol
Geometry
>>> Geometry <<<
IA
IC
CA
CC
MA
MC
FA
FC
XT
This geometry was written out using the "save" button.
The interactive geometry browsing as in VP1 is of course always active.
Rotz RotY
Zoom
Successfully loaded libGXGeometryPlugin.so
providing channels: Geometry
[Geometry/Geo] VP1GeometrySystem::buildController

In this session Andrea reads a file that was *not* produced by Athena…



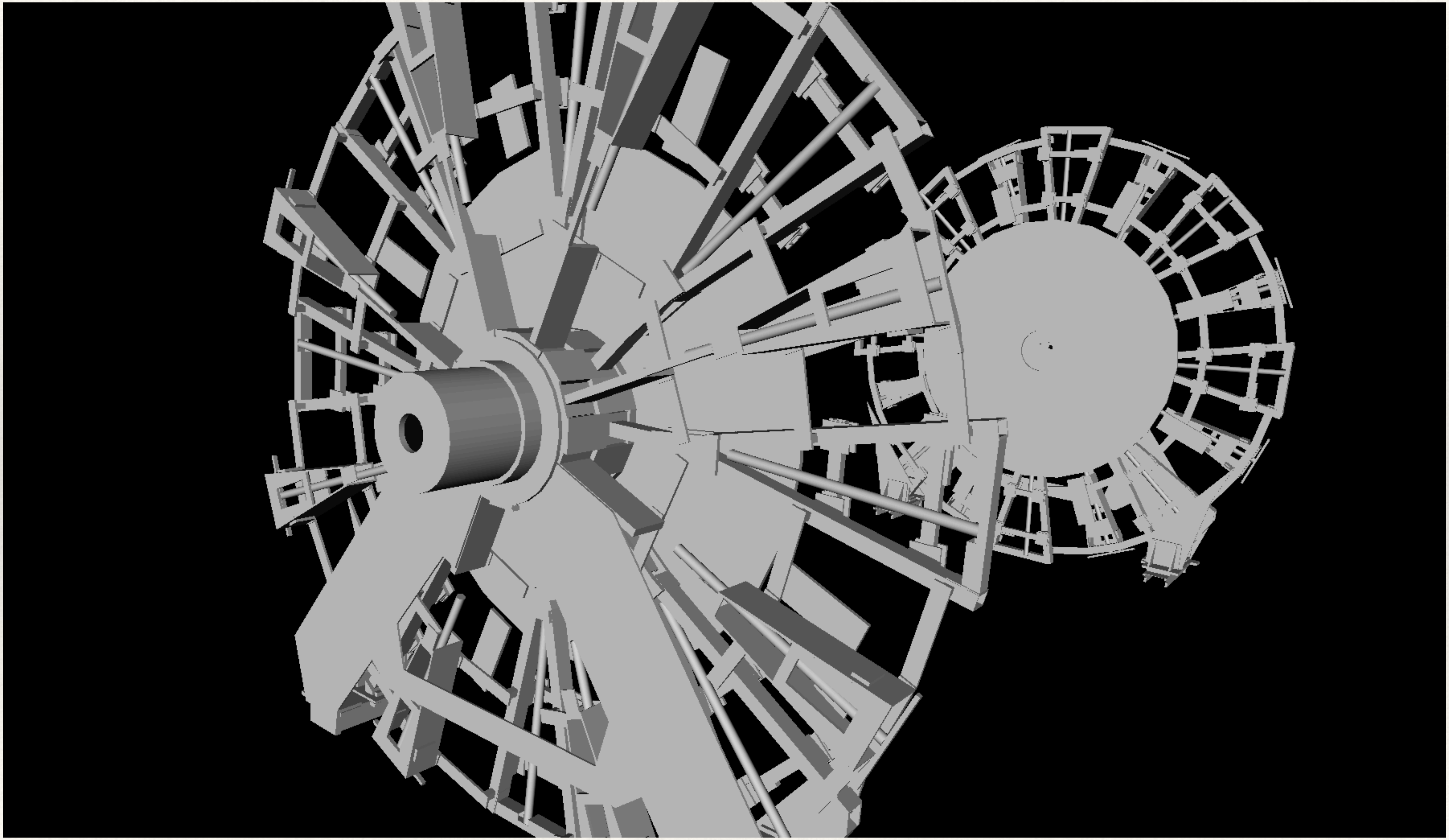
The file NSW.db came into existence as a proof of the principle that geometry can be developed on a laptop:

DD code building muon geometry was extracted from Athena, compiled standalone, and executed. The standalone program creates an output file* which is read by the Geometry explorer

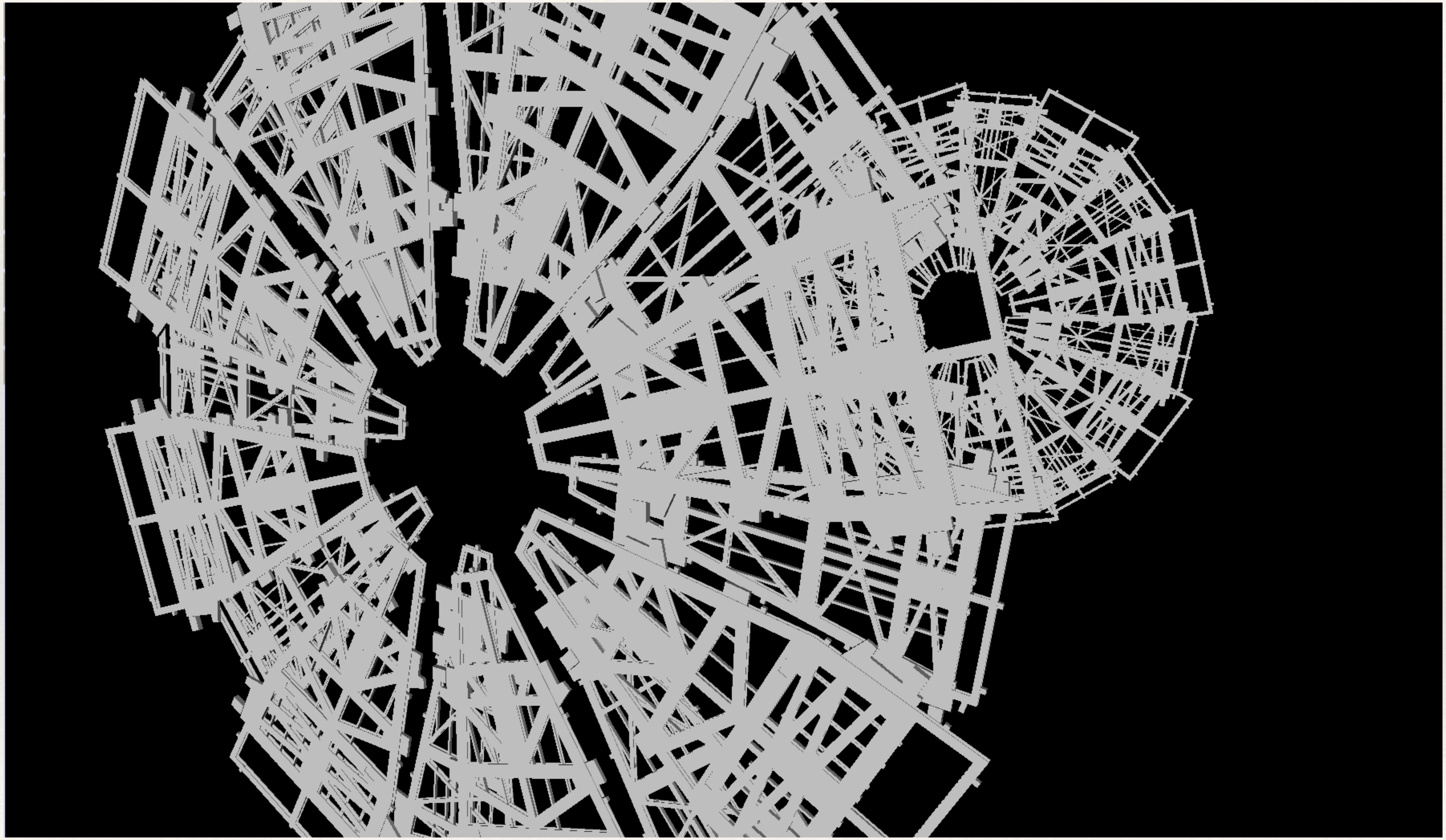


* in the future dual-use factories will allow to skip the intermediate step of writing and re-reading a file.

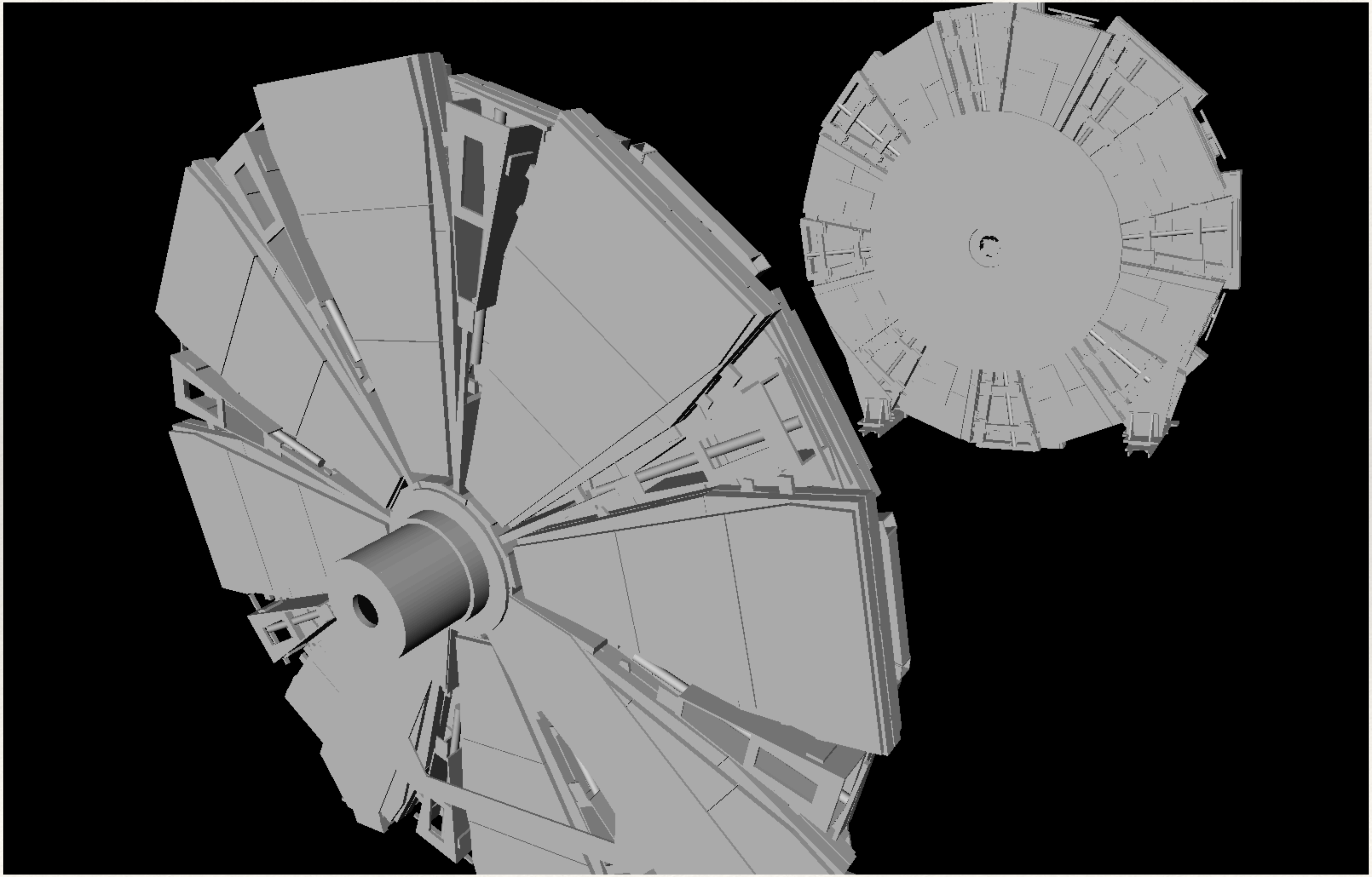
New Small Wheel (NSW) supports



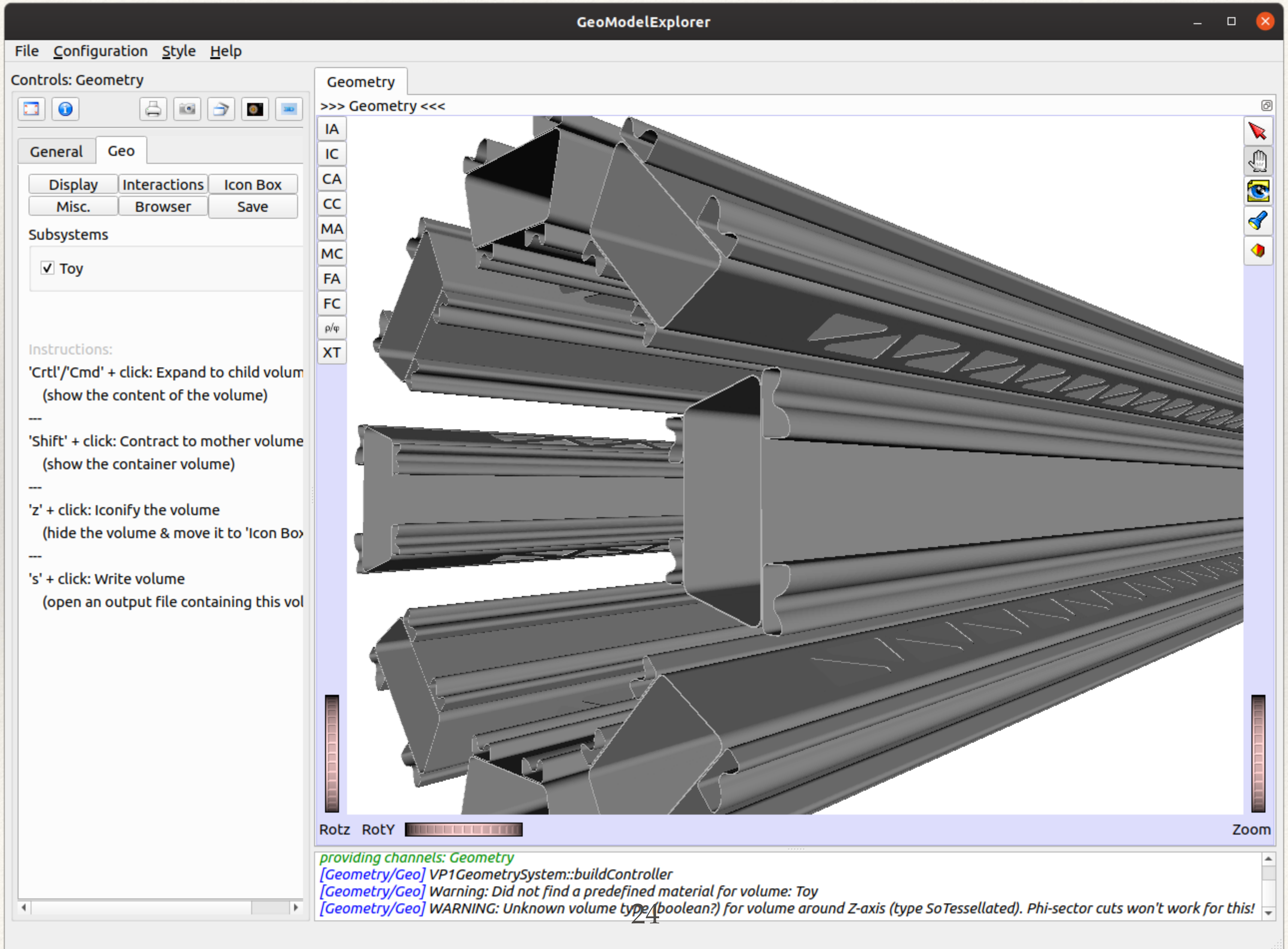
New Small Wheel spacers



The whole New Small Wheel



Importation of CAD files has also been demonstrated

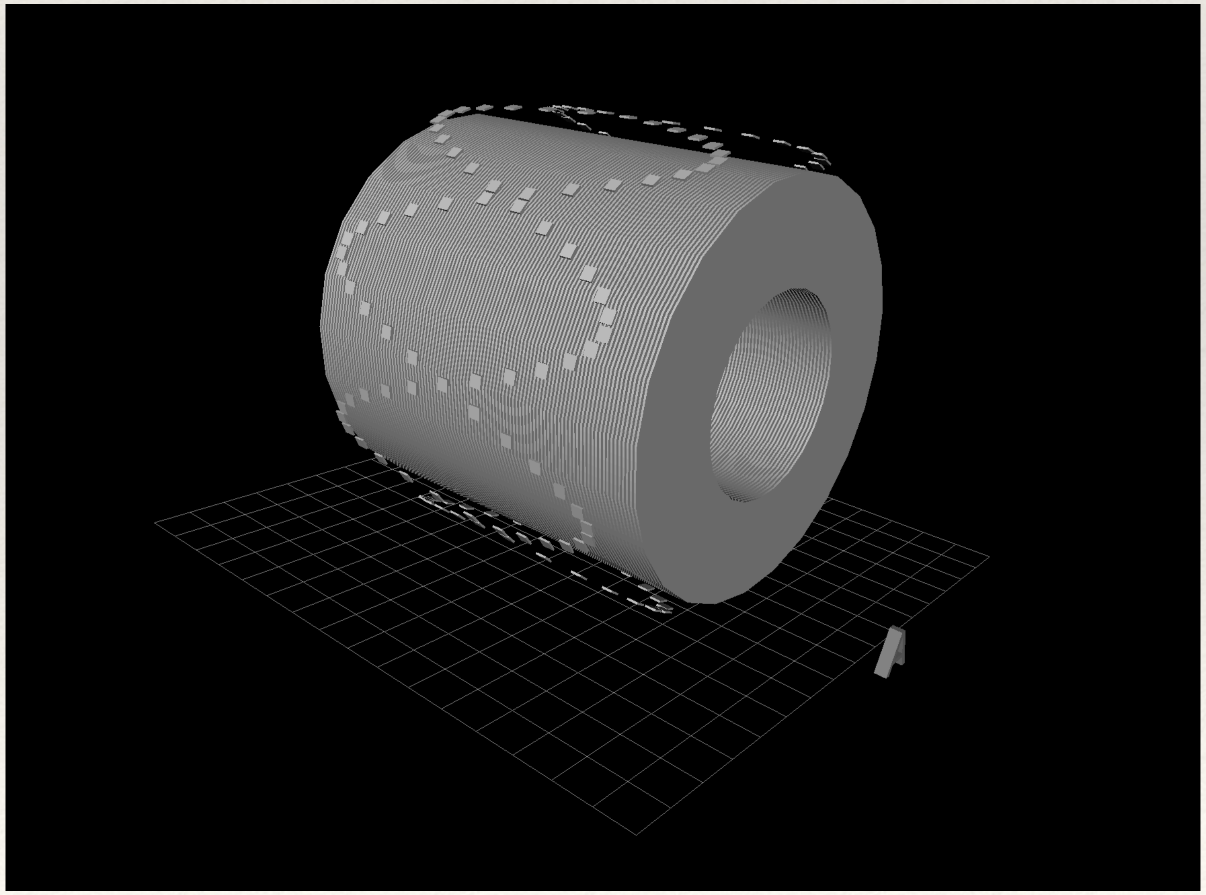


You can try this too!

Start with some of the example code in this repository:

https://gitlab.cern.ch/GeoModelDev/GeoModelExamples

HelloToyDetectorFactory ==>



In general terms:

There is now a tool that one can use to work on detector code without depending on Athena, lxplus etc. and which allows rapid turn-around times .

From the DD viewpoint, we have a proof of principle that the model we have in mind is workable (with a little help from our detector friends)

We will improve our tool on a release cycle in sync with ATLAS software weeks.

The major new feature we need is factory plugins, so we skip the intermediate step of writing/re-reading files:

```
gmex nsw.so trt.so …
```

When the new tools (chiefly database, geometry persistification, and visualization) have been deployed we will launch a significant effort to improve the detector description throughout ATLAS oriented towards successful Run IV operation (and where applicable to Run III).