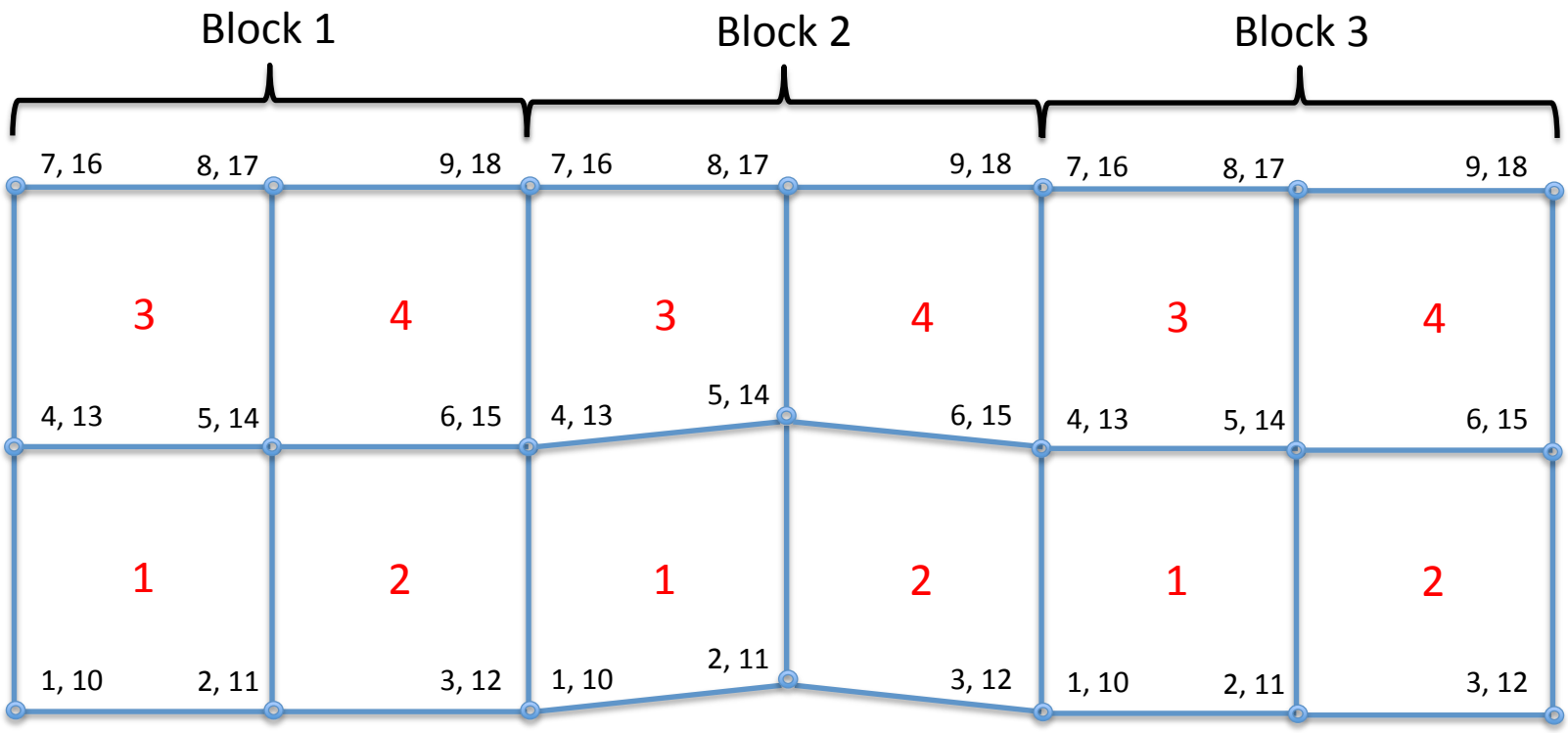
Making CGNS a little more usable for multi-block unstructured grids

Micah Howard

Srini Arunajatesan

Example multi-block unstructured grid



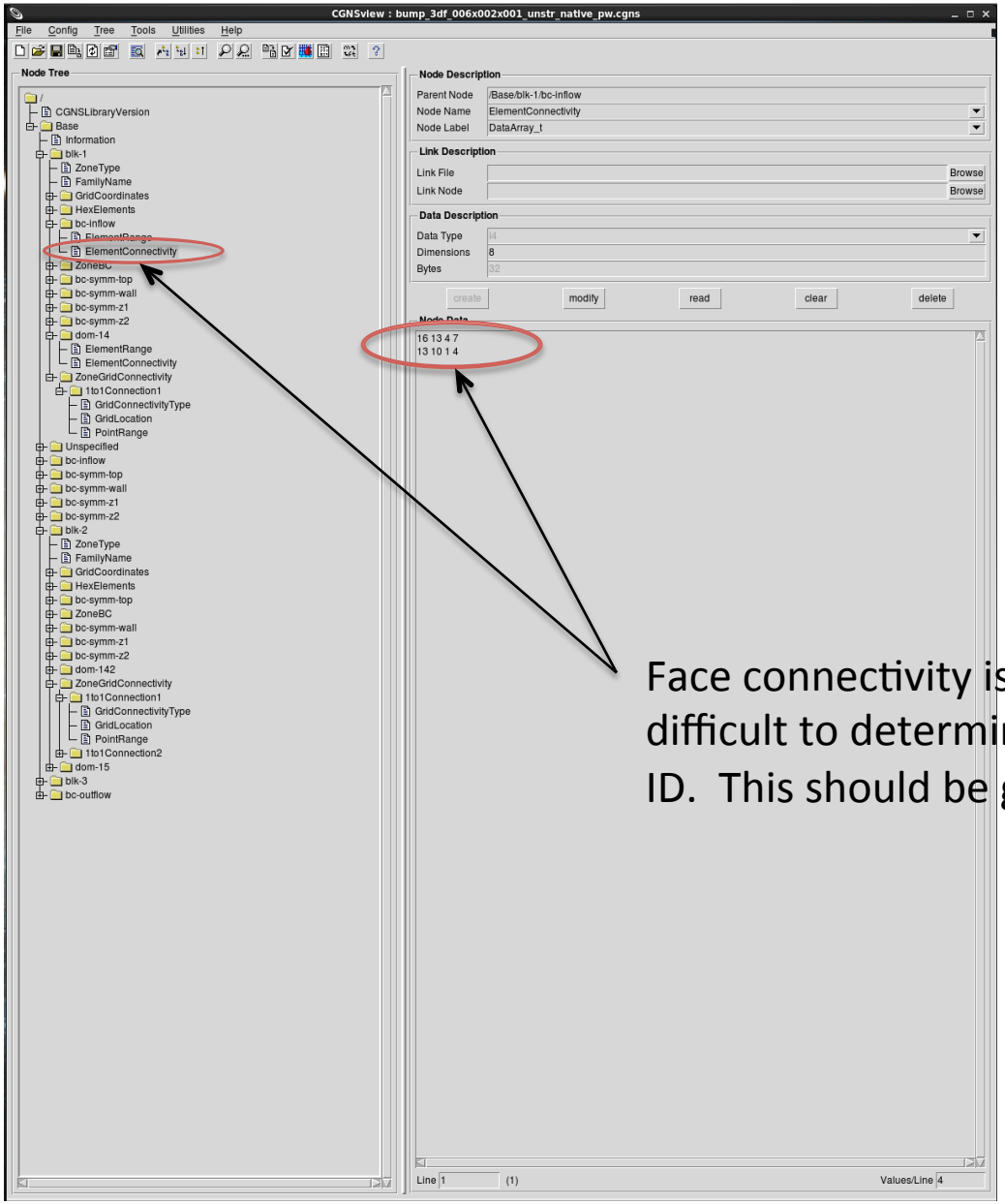
Including two optional pieces of information in an unstructured CGNS file makes it easier for the user…

- CGNS allows for this additional information to be included
- However, it doesn't seem to be standard practice for unstructured grids (based on what we've seen included in the CGNS file written out by a few different grid generators)
- It may be beneficial to have a "best practices" document that outlines what information is best to include for (multi-block) unstructured grids

What you get with in an unstructured CGNS file currently…

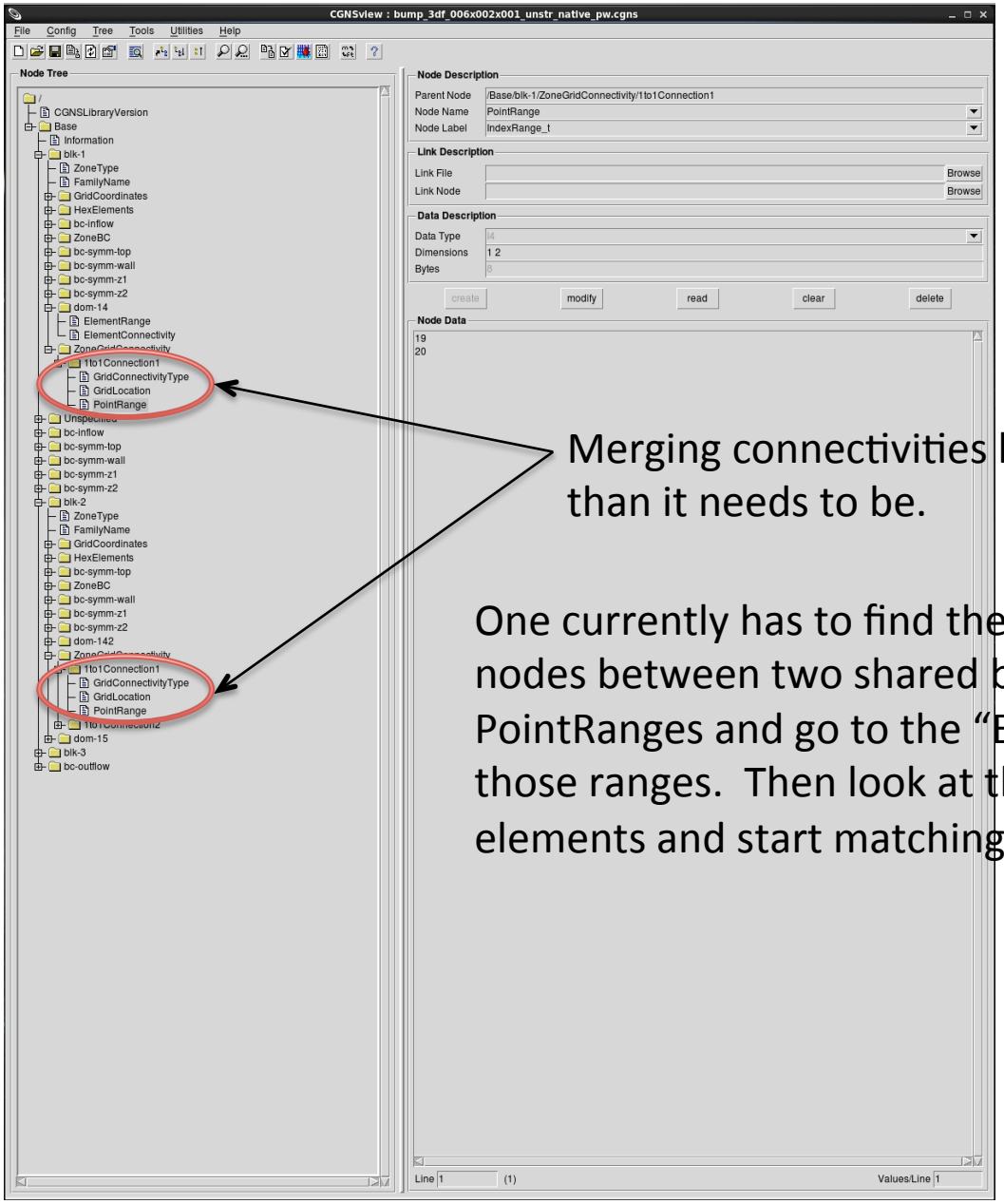
- The following CGNS file was generated with Pointwise
  - Sorry to pick on Pointwise but it's what we primarily use and seems to have the best CGNS support for unstructured grids (that we know of at least)

What you get with an unstructured CGNS file currently…



Face connectivity is given but it is very difficult to determine the parent element ID. This should be given.

What you get with an unstructured CGNS file currently…



Merging connectivities between blocks is more difficult than it needs to be.

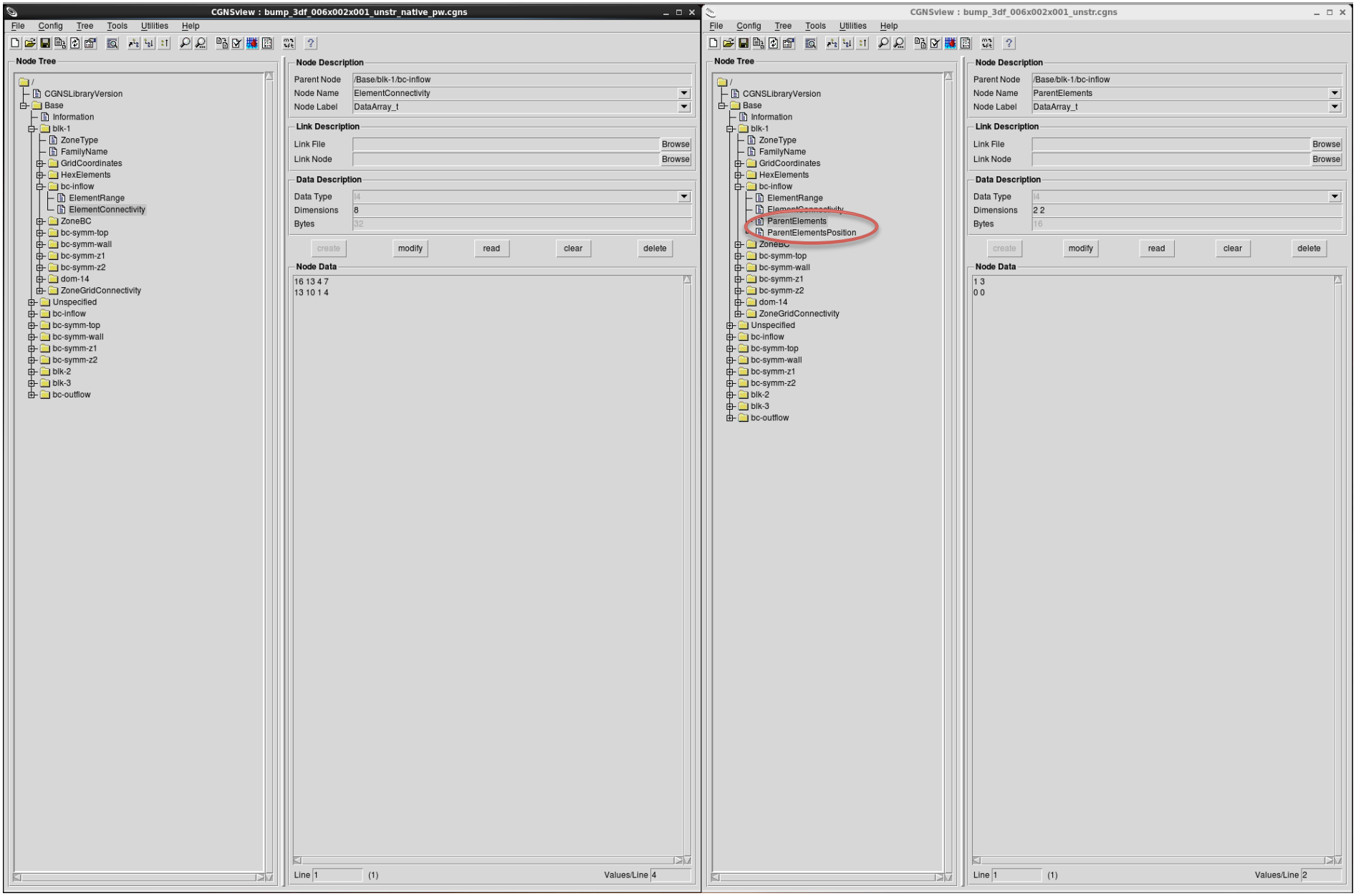
One currently has to find the matching “GridConnectivity_t” nodes between two shared blocks. Then look at the PointRanges and go to the “Elements_t” nodes that matches those ranges. Then look at the face connectivities of those elements and start matching nodes.

Including two optional pieces of information in an unstructured CGNS file makes it easier for the user…

1. Adding parent element data to the boundary “Elements_t” nodes in an unstructured block
   - This information provides both the “parent” element number and a means to get the connectivity of the face
   - Knowing the “parent” element number of a boundary face (without having to search for it) is very convenient for FE and cell-centered FV codes

This additional information will be shown on the following slides in comparison to a CGNS file as written by Pointwise

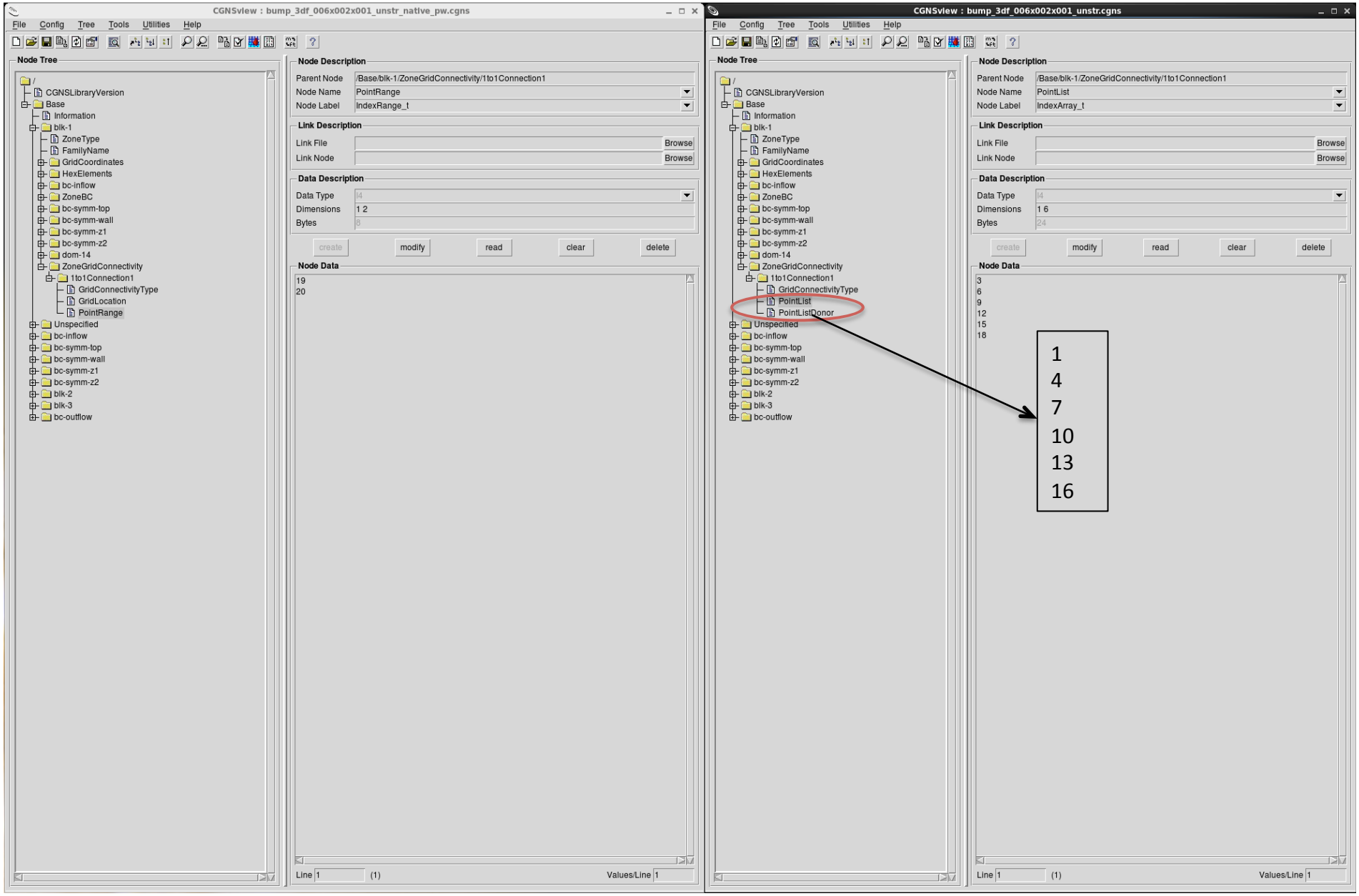
Parent element data



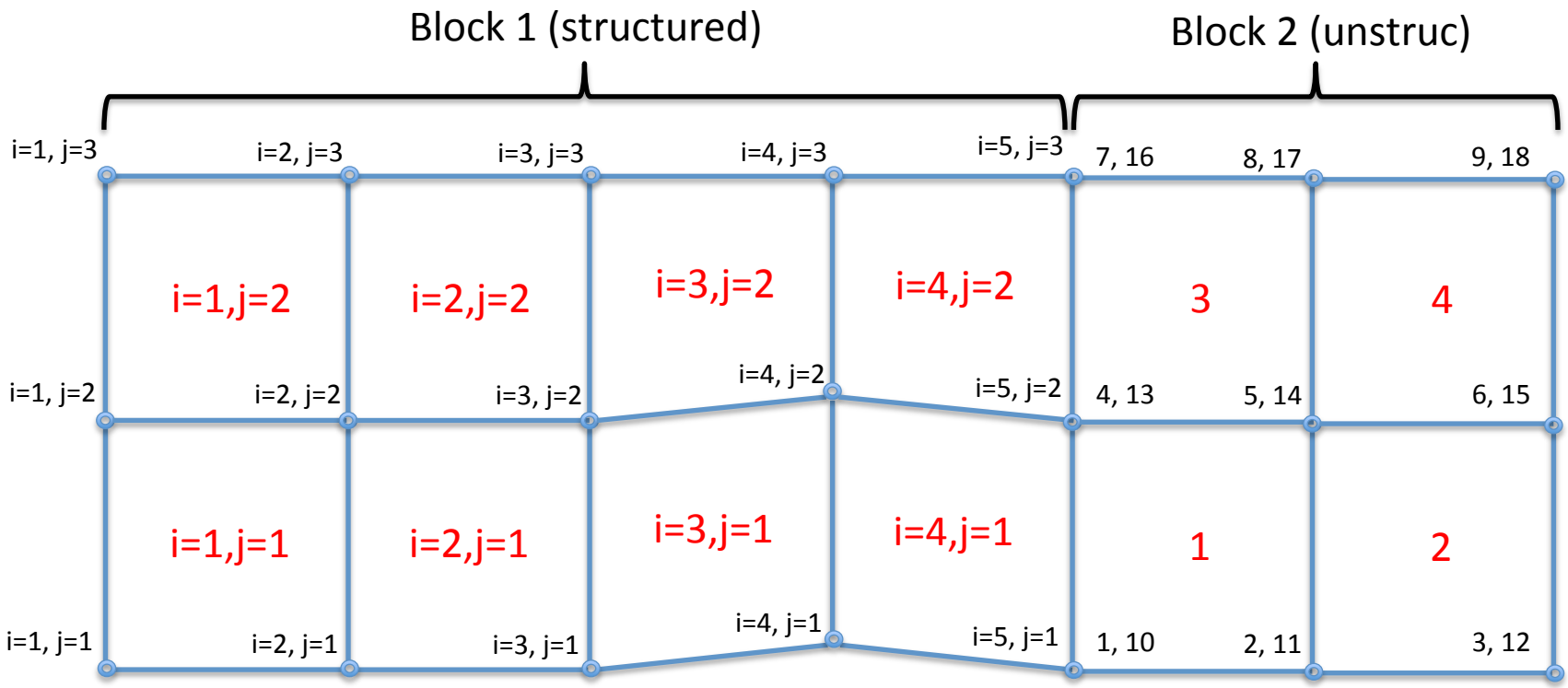
Including two optional pieces of information in an unstructured CGNS file makes it easier for the user…

2. Providing “PointList” and “PointListDonor” information in the “GridConnectivity_t” nodes of the “ZoneGridConnectivity_t”
   - This information provides a more direct path to “merging connectivity” between unstructured element blocks
   - If a user assigns unique node ids starting with the first block and working towards the last and uses the “PointList” and “PointListDonor” information to merge nodes, it is straightforward to get a consistent connectivity for all blocks

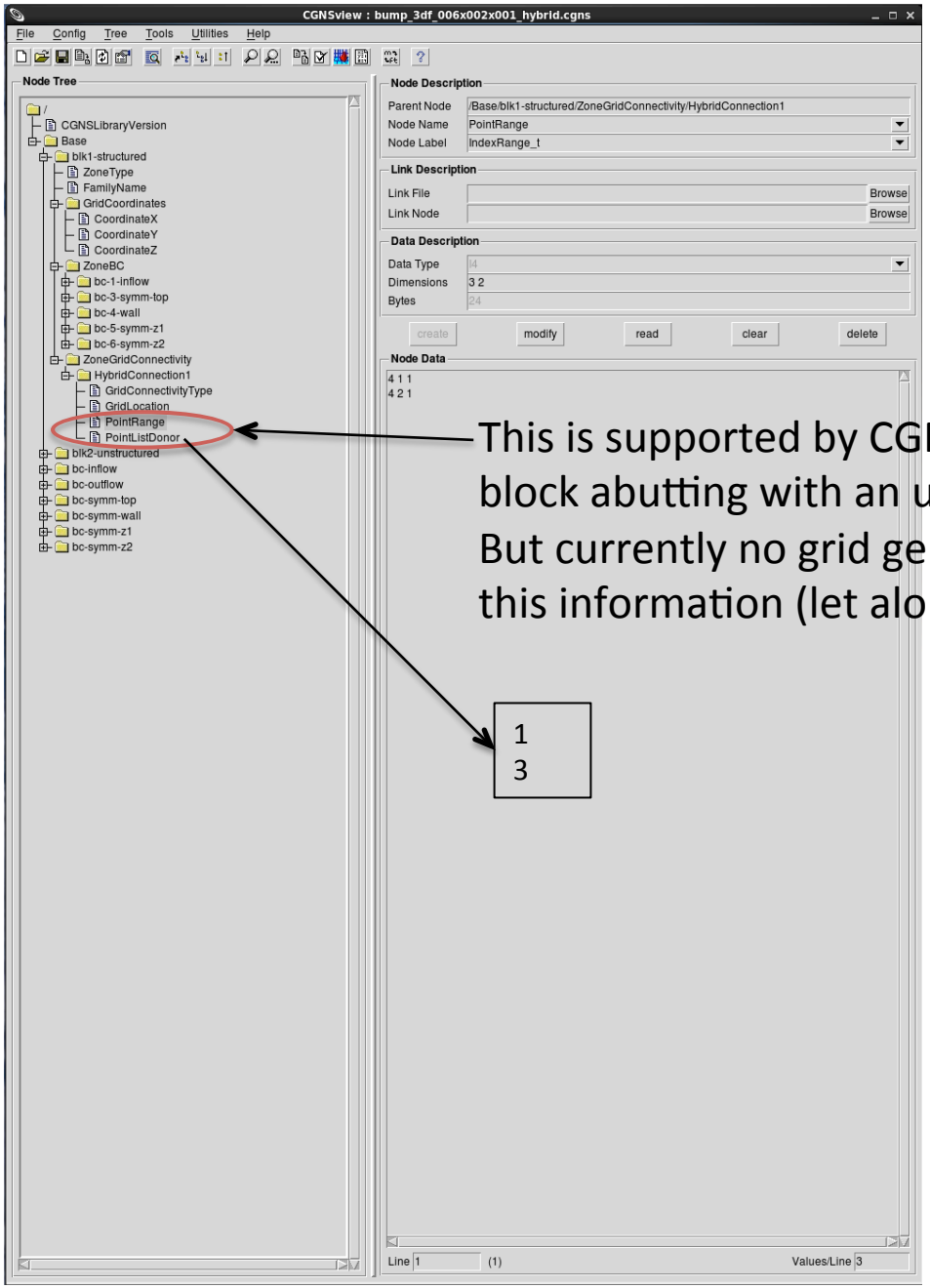
PointList and PointListDonor grid connectivity



Example hybrid grid

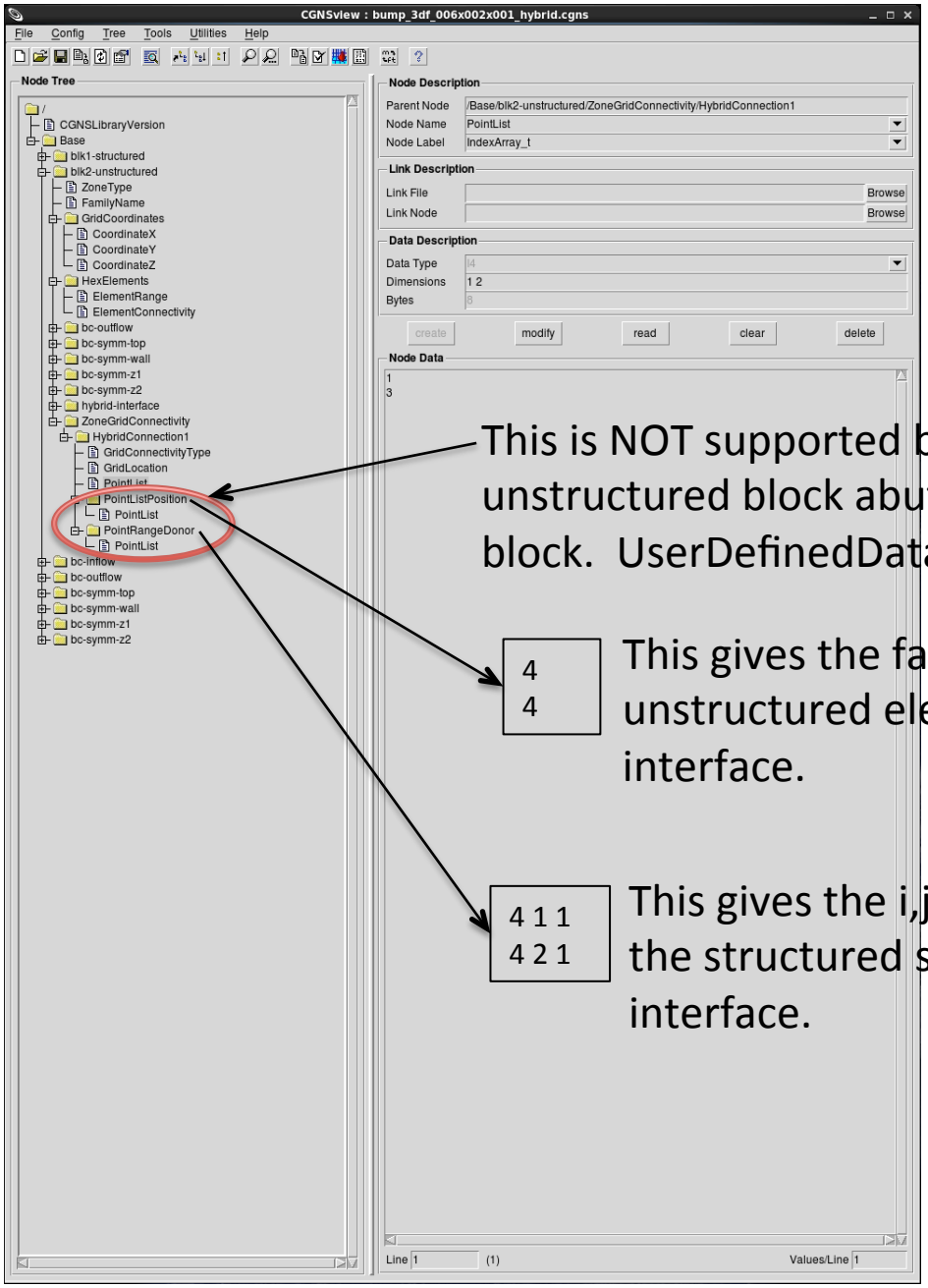


Hybrid GridConnectivity_t (structured->unstructured)



This is supported by CGNS for a structured block abutting with an unstructured block. But currently no grid generator will write this information (let alone a hybrid CGNS grid)

Hybrid GridConnectivity_t (unstructured->structured)



This is NOT supported by CGNS for an unstructured block abutting with a structured block. UserDefinedData is being used here.

4
4

This gives the face position of the unstructured element on the hybrid interface.

4 1 1
4 2 1

This gives the i,j,k range of cells on the structured side of the hybrid interface.