



# VizSchema – An Approach for Attribution of Data

Sveta Shasharina\*, John Cary, Marc Durant,  
Scott Kruger, Seth Veitzer

\* [sveta@txcorp.com](mailto:sveta@txcorp.com)

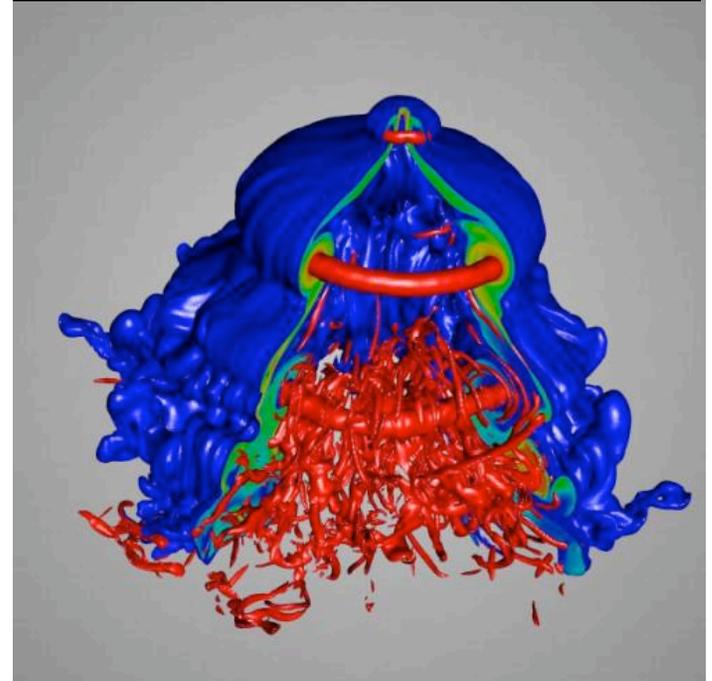
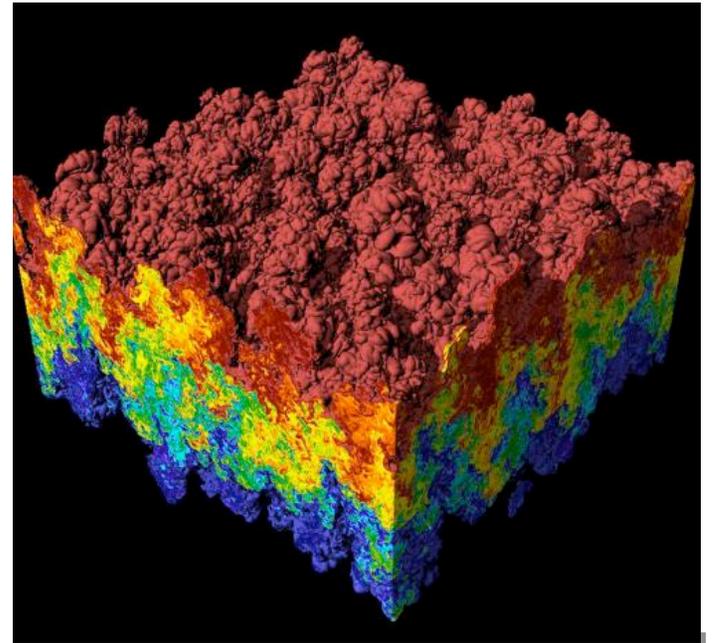


# Simulation Data and VizTools are Diverse

- Data
  - ASCII
  - All kinds of home grown binary format
  - Self-described data formats
    - NetCDF
    - HDF5 (our current choice)
- VizTools
  - IDL
  - Matlab
  - AVS/Express
  - Lately moving to open standard tools
    - VisIt (our current choice)
    - ParaView



- VisIt – supported by VACET SciDAC
- One needs to write a **reader** for each type of data format to transform into what VisIt expects





# One Cannot Write One Reader For All HDF5 data

- HDF5 consists of groups (like directories), datasets (like files – end leaves) and attributes (for small data and metadata)
- One can organize data in many ways using HDF5 constructs
- One could use any kinds of names and no metadata at all
- How one can understand what is what?
  - What is supposed to be visualized?
  - What order is used (row-major etc)?
  - Where is the mesh of the data?
  - What are the components of the mesh?



## I Hate ...

- writing readers
- undocumented data as much as I hate undocumented code



# VizSchema is an Attribution Standard

- Based on experience working with
  - VORPAL (uniform and unstructured meshes and particles)
  - NIMROD (structured meshes)
  - UEDGE (structured, multi-domain meshes)
  - TXFLUID (unstructured meshes)
  - FACETS (combination of the above)
- VizSchema
  - Data Model (agreement about data organization and metadata)
  - C++ reader (independent of Viz tool) of HDF5 data into in memory viz objects
  - VisIt plugin
- Funded by FSML grant (DOE SBIR, FES), FACETS grant (DOE SciDAC) and Tech-X Corporation



# VizSchema: Minimalistic but Enough for Viz and Analysis

- Viz entities (based on our experience):
  - Variables (live on external mesh)
  - Variables with meshes (spatial info is mixed in)
  - Meshes
  - Derived variables
- Metadata
  - Identifying the entities and specifying their kinds (if any)
  - Providing information needed for minimal viz
- Principles
  - Metadata is minimal
  - Metadata is in attributes starting with “vs”
  - Groups and datasets names are not regulated



# Variables Need to State Type, Centering (and Ordering)

```
Dataset "phi" {  
  Att vsType = "variable" // Register as something  
    // to "look at" and look for vsMesh  
  Att vsMesh = "mycartgrid" // This should exist  
    //somewhere in the file  
  Att vsCentering = "zonal" // Nodal is default  
}
```



# Variables With Mesh Show Where to Find Coordinates

```
Dataset "vorpalelectrons" {  
  Att vsType = "variableWithMesh" // Find  
    //vsNumSpatialDims or VsSpatialIndices  
  Att vsNumSpatialDims = 3 // First 3 indices are  
    // coordinates  
  Att vsIndexOrder = "compMinor"  
}
```

```
Dataset "synergiaElectrons" {  
  Att vsType = "variableWithMesh"  
  Att vsSpatialIndices = [0, 2, 4] // These indices  
    // are coordinates  
}
```



# Meshes Specify Kinds And Things to Build Itself

- Structured grid has just a list of points
  - Need vsOrder (if not default)
- Rectilinear mesh has list of numbers in each direction
- Uniform mesh has number of points in each direction, max and min in each direction
- Unstructured mesh needs points and elements (tris, quads, tets, hexs, prisms, etc)
- The list of meshes will grow
- Work with users to provide alternative ways to describes same kinds of meshes



# Derived Variables

```
Group anygroupname {  
    ATT vsType = "variableDefinition"  
    ATT vsDefinition = "elecEnergyDensity =  
(E_0*E_0+E_1*E_1+E_2*E_2)"  
}
```

- Count on expression language of VisIt
- Use variables as basis
- Make viz richer

# How To Adopt VizSchema?

- Change your I/O
  - VORPAL
  - FACETS (Fusion SciDAC)
  - NIMROD (MHD)
  - PolySwift++ (nanotech)
- Or change your files using PyTables (very easy interface to modify and add attributes)

```
h5file = tables.openFile(fileName, mode='a')
dataSet = h5file.getNode("/") + dataSetName)
dataSet.attrs.vsType = "variable"
h5file.close()
```

- Changing old VORPAL outputs to fit evolving schema
  - Changing SYNERGIA output
- First step is look at “h5dump -A” command and send to us...



## Future Directions

- Finish parallel plugin (started by G. Weber)
- More codes and updating schema as needed
- VORPAL and FACETS skin
- NetCDF?
- Auto-generation of correct markup from a text or XML document



## Information

- VizSchema information and download:  
<https://ice.txcorp.com/trac/vizschema/wiki/WikiStart>
- One of standard VisIt plugins
- sveta@txcorp.com



## Thanks to:

- VisIt team (H. Childs, S. Ahern, J. Meredith, B. Whitlock, A. Sanderson)
- Tech-X FACETS and VORPAL teams
- C. Geddes and G. Weber