A quick tour

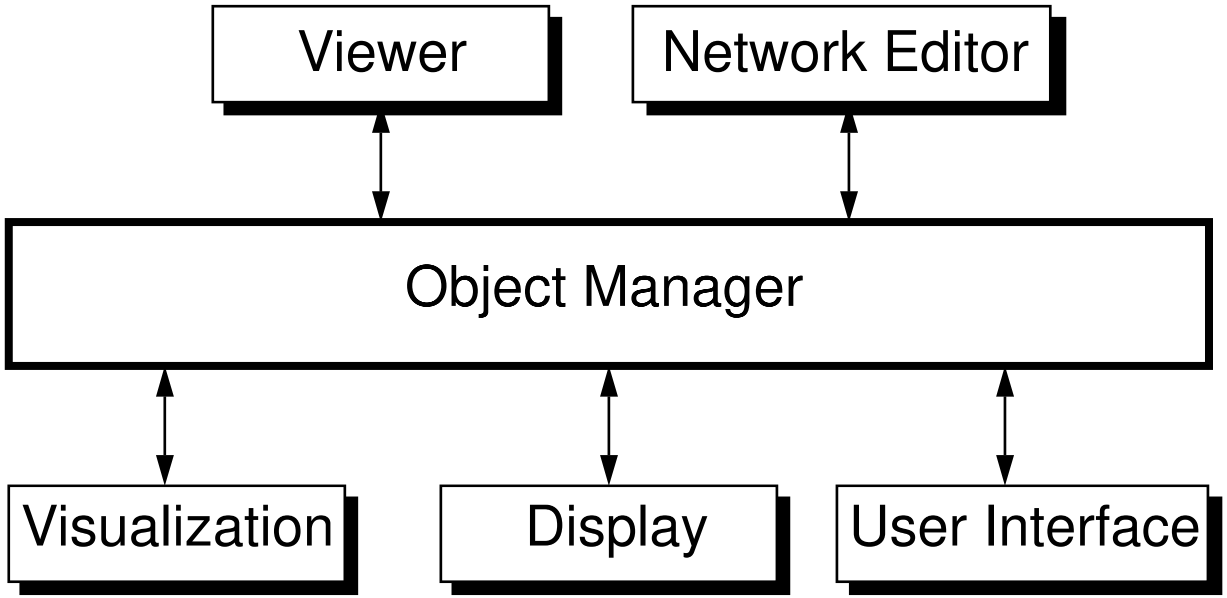
- Architecture overview
- Network editor
- Data model
- Data Viewer
- Visualization techniques
- Handling large datasets
- User interface
- PC Support

Developers and Visualization Edition

- There are two different editions of AVS/Express:
  - Visualization Edition - upgrade to AVS5
  - Developers Edition - producing custom applications within your organisation or for external customers
- Both have the same architecture
- We will discuss the differences later
- Throughout the tutorial we will refer to AVS/Express - Visualization Edition as “Viz/Express”

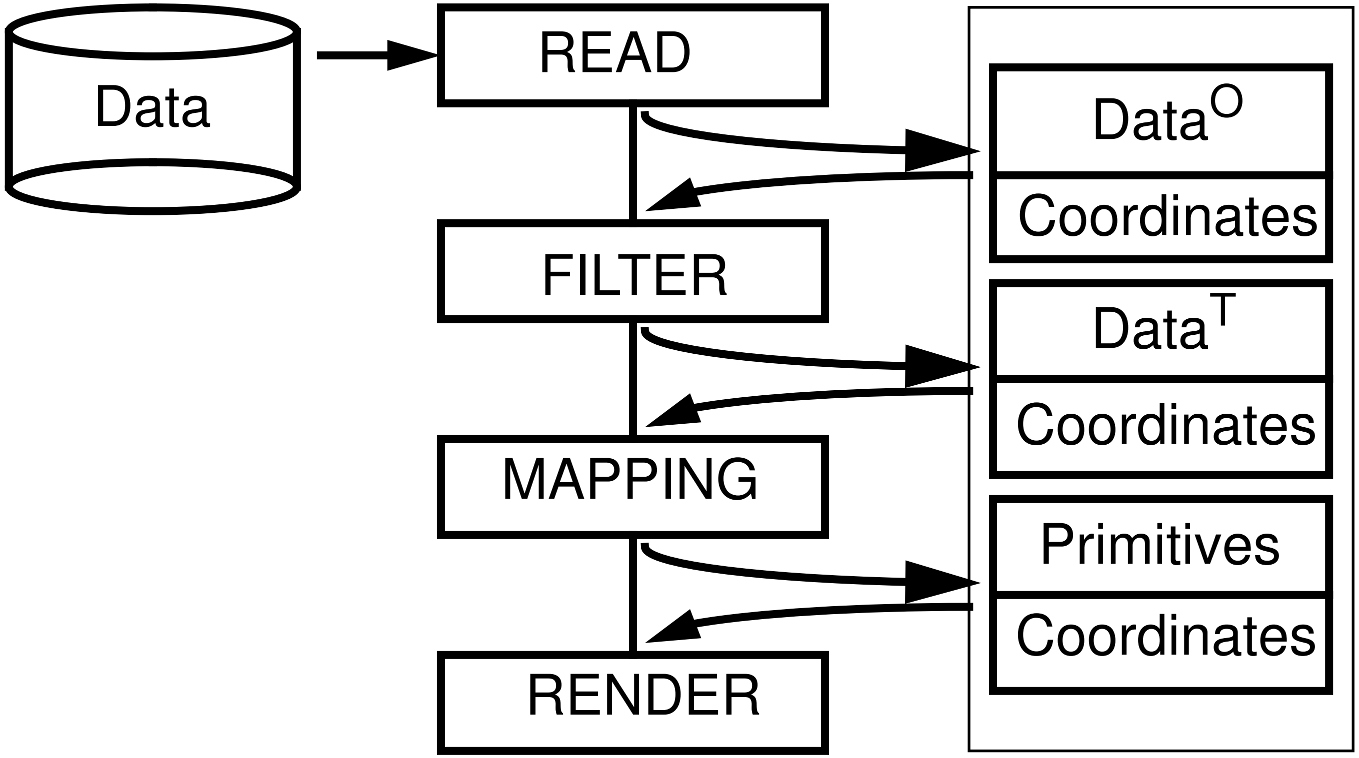
Architecture overview

❑ It is an object-oriented, visual development tool that enables you to build reusable application components



Next Generation Systems

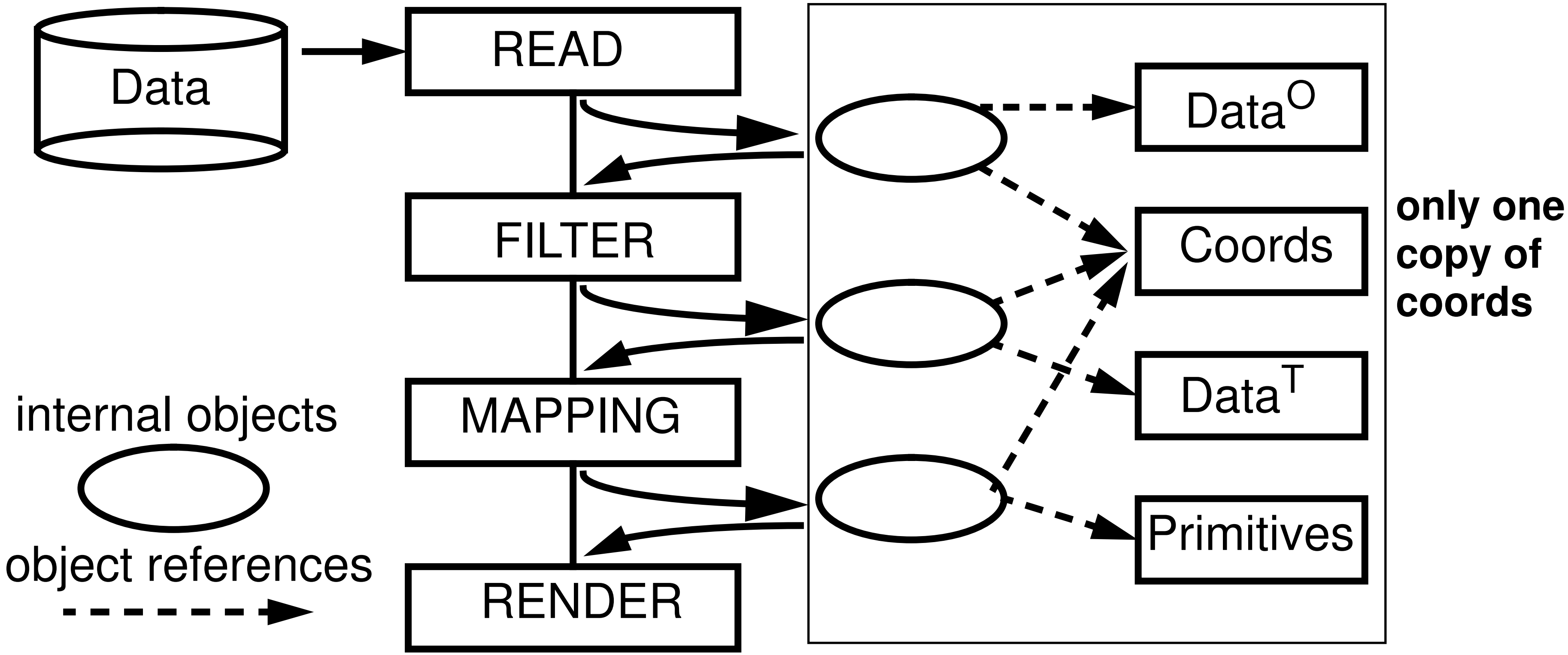
- ❑ Dataflow paradigm seen as too restrictive



3 copies
of the data
and coords

Object Manager

- ❑ notion of a central object store

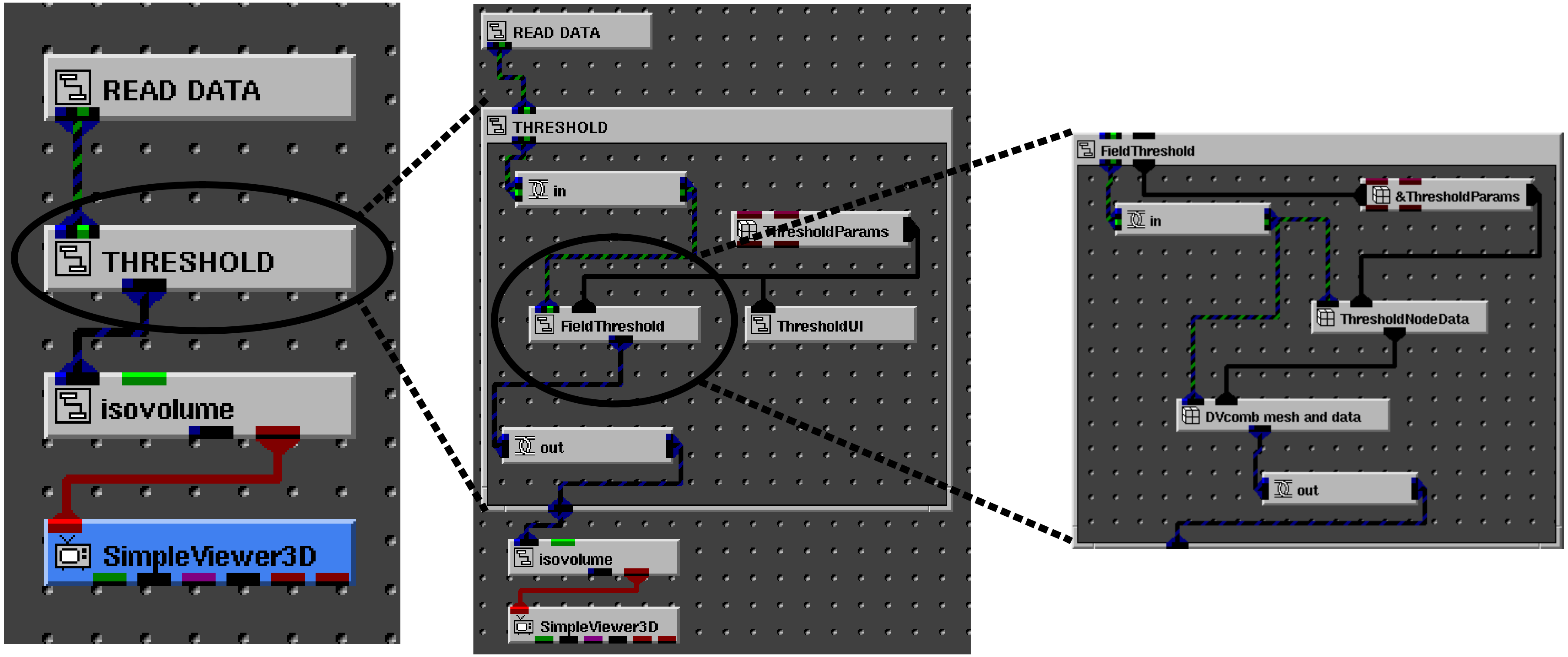


Network editor

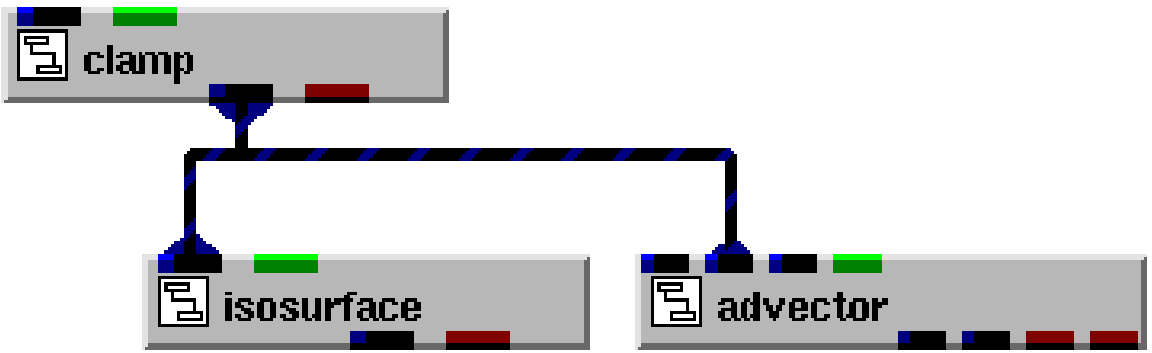
- This has been extended to support both coarse and fine grain visual programming
  - build applications
  - integrate code into the system
  - create data structures
- It can be used to define and display:
  - data dependencies between compute modules
  - data references
  - parent/child relationships

Visual Programming

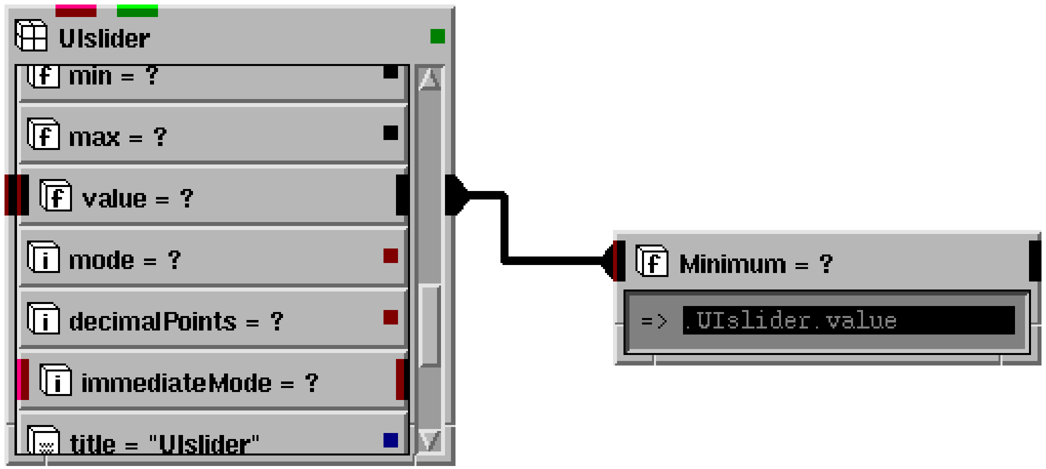
- Taken a step further where the whole module and its components are developed using visual programming



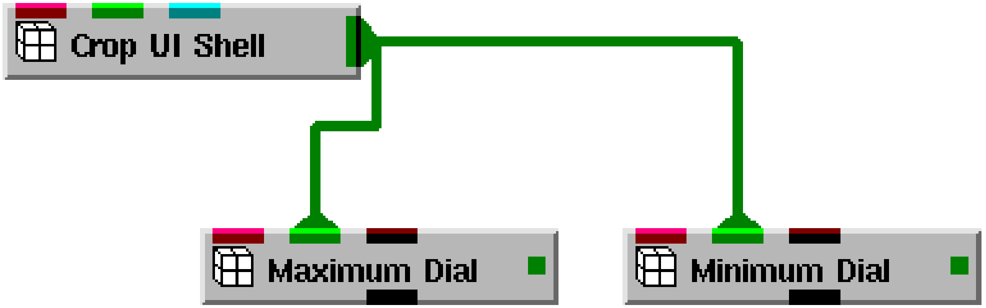
Data dependencies



Data references



Parent/child relationships



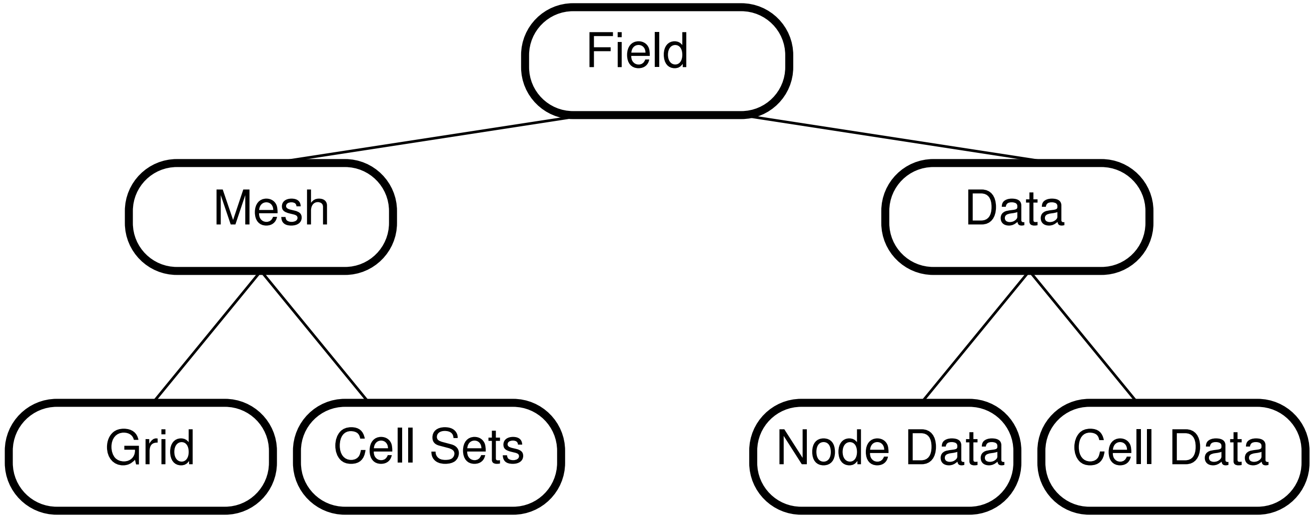
Demonstration

- Starting Viz/Express, tour of libraries, how to search, what does () mean, where's the hammer gone? Online help
- Build Slice and Isosurface application
- Data viewer functionality
- Different objects...
- library - collection of objects
- macro - combination of modules, macros, connections
- module - parameters and methods that interface to application code
- parameters - holds data which is operated on by methods
- methods - interface to application code
- links - route connections in/out of macro or application object

Data model

- ❑ Now just one unified data model called a “Field”.
- ❑ It replaces
  - AVS5 field
  - Unstructured Cell Data (UCD)
  - AVS5 Geometry
- ❑ Other types are handled differently:
  - colourmaps
  - Chemistry Developers Toolkit (CDK)
  - user defined data

Structure of the Express Field



- Geometry is treated as a subset of unstructured data
- Points, lines, triangles and polygons are different cell types
- We will deal more with data model in “Importing Data”

Data Viewer

- Unified approach to rendering text, images, 2D and 3D geometry and volumes
- Mixed mode rendering allows different primitives to be shown in the same view e.g.,
  - volumes with embedded geometry - probing a volume
  - image in the background of a 3D geometric scene
- 2D rendering implemented on 2D libraries
  - X on Unix workstations
  - GDI on Windows PCs

3D Rendering

- 3D rendering using OpenGL, PEXlib and XGL
- Implementations usingPEXlib and OpenGL will support distributed graphics rendering
- Data Viewer no longer part of AVS kernel but a separate object

Visualization techniques

- All the AVS5 techniques are included (under some new names)
- Due to unified data model there is only one Isosurface now needed
- Most modules operate on any primitive data type (byte, short, int, float, double) and 1,2 and 3D data
  - no need for field_to_XXXX modules
- There are many new modules as well...

New modules

- axis2D, axis3D
- contour
- cut
- cell_data_math
- data_math
- external edges
- external faces
- geo_glyph
- glyph
- interp_data, interp_cell_data

More new modules

- interp_texture
- isoline
- isoline_trace
- isosurface_trace
- iso_probe
- isovolume
- mirror
- netCDF read and write
- normalize
- plane_crop

Even more new modules

- probe cell
- reset_xform
- scale
- scat_to_unif
- select_cells
- set_null
- set_xform
- slice_texture2D, slice_texture3D
- solid_contours, surface_plot, texture_mesh, thresh_cell, thresh_null

Visualization techniques

- Cylindrical/spherical/polar coordinates and NULL data are supported
- 2D graphing - integration of some Toolmaster components
  - curve, bar, barline, area, staircase
  - 2D isolines, filled contours, data mapped isolines
- Hardcopy support for Postscript and CGM
  - full device resolution output
  - 2D text annotation using Postscript scalable and variable fonts
  - 3D geometry rendering upto 300 dpi

The V language

- fundamental to the use of AVS/Express
- many operations can however be carried out in a visual mode by the network editor
- V replaces CLI

Purpose of V

- saves the state of AVS/Express
  - object definitions, templates and active instanced objects
  - connections
  - parameters and their values
- describes the structure of modules (see later)
- used as a scripting language
- an interactive tool to manipulate and debug the AVS/Express state

How can you use V?

- typing values into VCP (command line processor)
- reading a file containing V code
- issue commands an API
  - inside a modules method
  - outside the system through a socket connection when in server mode

V statements and commands

- V consists of
  - commands - begin with a dollar ($)
  - statements
- Statements perform threee actions:
  - create
  - modify
  - delete

Examples

```
MODS.Read_Field Read_MyData;
MODS.isosurface isosurface {
in_field=> <-.Read_MyData.field
};


group Adder {
int a;
int b;
int res => (a + b);
};
```

Handling large datasets

- Alternate object rendering: allows you to specify a secondary object
  - wireframe of a geometry
  - sub-sampled image
- Dynamic objects: small objects can be rendered using non-invasive drawing mode
  - probing a large volume
- Direct rendering
  - rendering without the need to generate intermediate in-memory copies of the resulting geometry

User interface

- AVS/Express is no longer based on LUI
- Motif on Unix and Windows on PC's

- You can build user interfaces using the Network editor
- This will be shown later...

PC Support

- AVS/Express Release 3.0 requires an Intel 486 or Pentium system with a colour monitor
- It is not recommended to use:
  - 486sx (no math coprocessor)
  - 386
- Recommend a minimum of 32MB of real memory
- AVS/Express on Microsoft Windows NT: 3.51 or a later version
- AVS/Express on Microsoft Windows 95: official released version
- Supports the sw renderer and the OpenGL hw renderer (WinGL)

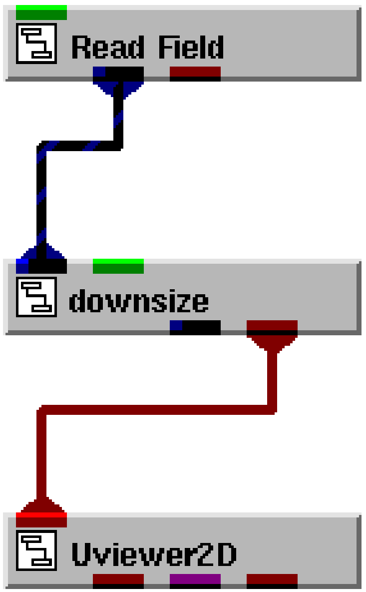
Developers and Visualization Edition

- In Viz/Express you CAN write modules
- Both based on same architecture...
- AVS/Express Developers edition provides:
  - more of underlying system exposed
  - low level objects which are used to build high-level modules
  - API's to each component
  - Graphics Display Kit - ability to construct a custom viewer
- Produce a cut-down binary of your application for distribution

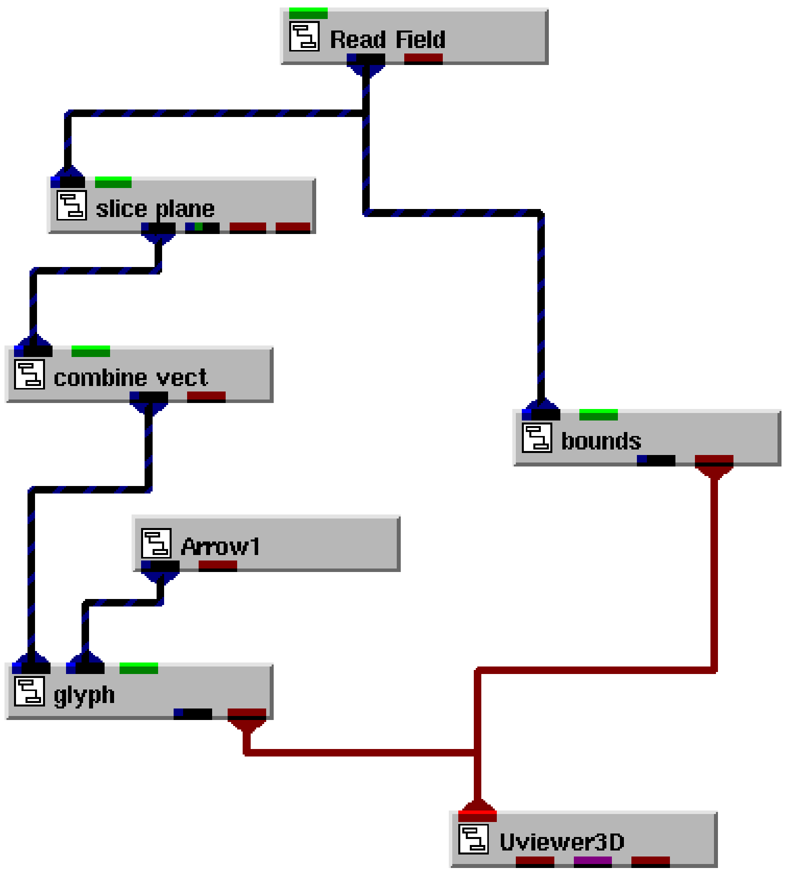
Demonstrations

- V language
- Difference between Developers and Visualization edition
- 2D and 3D examples...

2D example



3D example



Read Field
slice plane
combine vect
bounds
Arrow1
glyph
Uviewer3D