AVS/Express - an object-oriented visual development tool

Bruno Proix *Advanced Visual Systems, Clichy, France*

Abstract

This paper describes a new generation of software dedicated to the development of applications that require advanced 2D and 3D visualization techniques.

Keywords : AVS/Express, visualization, development, object-oriented, C++

1 Introduction

Using graphics to visualize data in order to understand it has always been a key issue for people like scientists that need to manipulate large amounts of data. The development of graphics hardware capabilities has led these people to look for new kind of graphics software, especially for 3D graphics.

That is the reason why, in the late ‘80s, software packages such as AVS, IRIS Explorer and Data Visualizer appear on the scene. The goal of these was to provide users with a tool to create complex visualization scenes without needing to program hundreds of lines of code. These new visualization products were very well suited for end-users that wanted a solution for visualizing their data. However, they were not well suited for developers that wished to build their own visualization applications that they could re-sell and distribute.

Some of the reasons for this were :

- Lack of configurability of the entire user interface of the products.
- Restrictive access to and manipulation of data.
- Efficiency problems due to a data flow paradigm.
- Difficult programming interface to manipulate geometries, particularly when upstream data was concerned.

So, in 1992, Advanced Visual Systems decided to start a project to design a new generation of software that could still be used by end-users to visualize their data in a fast and efficient way, but that also could be used by developers to build custom applications to distribute to end-users.

2 Why a new product ?

The characteristics of such a new product were defined as:

- Utilizing an object-oriented design for more flexibility and usability, including object re-use.
- Providing a very high level of access to the user interface portion of the product.
- Offering uniform access to high-performance rendering functionality on a wide variety of platforms, for text, images, 2D and 3D geometries, and volumes.
- Using state-of-the-art technology for the rendering pipeline.
- Providing a flexible and extensible data model.

- Using a data reference architecture, instead of data flow, for performance efficiency and flexibility.
- Providing support for large images and other large datasets.
- Providing interfaces to major databases.
- Offering automated cross-platform portability between all the major UNIX flavors as well as PC's.

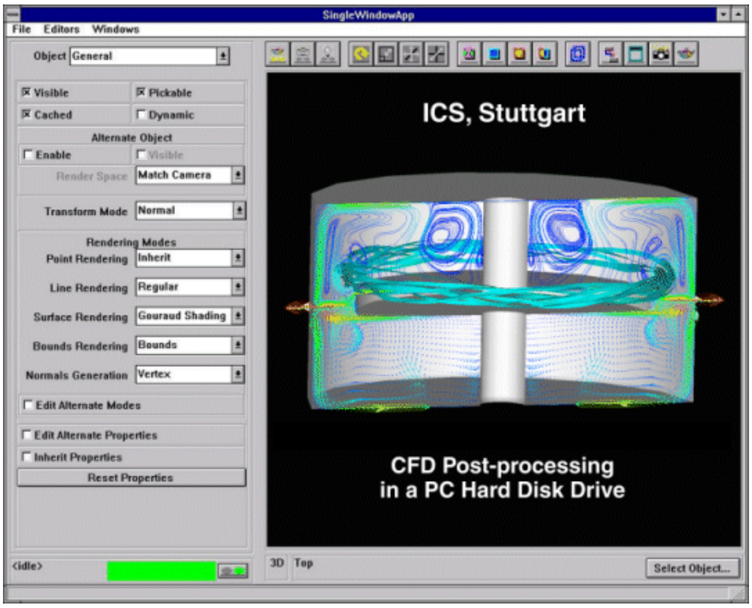
AVS/Express was developed to meet these requirements.

3 AVS/Express overview

AVS/Express is available in two versions :

- AVS/Express Visualization Edition
- AVS/Express Developer Edition

The main difference between these two versions is the ability for the Developer Edition to generate a runtime, stand-alone application that can be distributed to other people.



SingleWindowApp

Object: General

☒ Visible ☒ Pickable

☒ Cached □ Dynamic

Alternate Object

□ Enable □ Visible

Render Space: Match Camera

Transform Mode: Normal

Rendering Modes

Point Rendering: Inherit

Line Rendering: Regular

Surface Rendering: Gouraud Shading

Bounds Rendering: Bounds

Normals Generation: Vertex

□ Edit Alternate Modes

□ Edit Alternate Properties

□ Inherit Properties

<idle>

Figure 1 :CFD post-processing with AVS/Express. Courtesy of ICS - Stuttgart

Major features of AVS/Express include :

- advanced graphics tools,
- visual programming interface,
- cross-platform (UNIX / PC NT & Windows 95),
- code import (integration of external user code),
- component export (Dev Edition only) : C++ classes, OCX

4 A new architecture

Since AVS/Express is object-oriented, at its core are a number of objects. A central underlying element of the system, called the "Object Manager", manages the relationships between all of the objects used.

4.1 The Object Manager

The Object Manager is the central control mechanism in AVS/Express that manages the definition and execution of an application built with AVS/Express objects.The Object Manager defines data and functions (with interactive validation) during object creation, actually accesses the data, manages the connections among objects and schedules their execution, executes objects' methods, handles API calls to update objects and their data, and supports multi-process computing. The user interface to the Object Manager in AVS/Express is through:

- A command language called "V" (as in "Visual").
- A visual programming tool called the "Network Editor".
- Application interfaces (API's) for C, C++, FORTRAN, and OLE/ActiveX.

V is a declarative, ASCII-based language with a C-like syntax, that also includes a powerful mathematical expression evaluator.The data model used in AVS/Express for storing and accessing technical data offers unified support for structured, unstructured and geometric data.. This permits the use of the same modules for rendering a uniform dataset or an FEA-like structure. This flexibility illustrates the polymorphic nature (in object-oriented terms) of the data visualization techniques available in AVS/Express.It is also very easy to do define new data types or to enhance existing data types in AVS/Express.For instance, suppose you are building an application for medical imaging (e.g., the display of images coming from scanners). Your requirements include manipulating images for display and applying some image processing techniques, and you are using a data structure called "image". Perhaps you would like to associate some additional information with that data structure, for example, information related to the image such as patient name, image date, scanner type, etc.The syntax to acomplish this using "V" is the following:

```
image A;
custom_type B;
A+B custom_image;
```

This custom_image type can still be used with the modules taking ‘image’ type as input (like image processing modules), because the custom_image base type is ‘image’.

4.2 The Network Editor

Another way to manipulate objects in AVS/Express is through the use of a mouse-driven visual programming interface called the Network Editor.The Network Editor consists of a work area into which you can "drag and drop" objects from various libraries, and visually construct data types, modules and applications. Modules in the Network Editor offer simple, menu-driven interfaces to edit module properties, modify values or connect modules together.When you link two objects together, the color of the link actually provides a visual indication of the data types of the connected parameters.[1]

Users of the Developer Edition can "drill-down" into every module in the system to access the lowest levels of a module containing its parameters and methods, and can change the behaviour at any level of the module by simply modifying parameters' values.

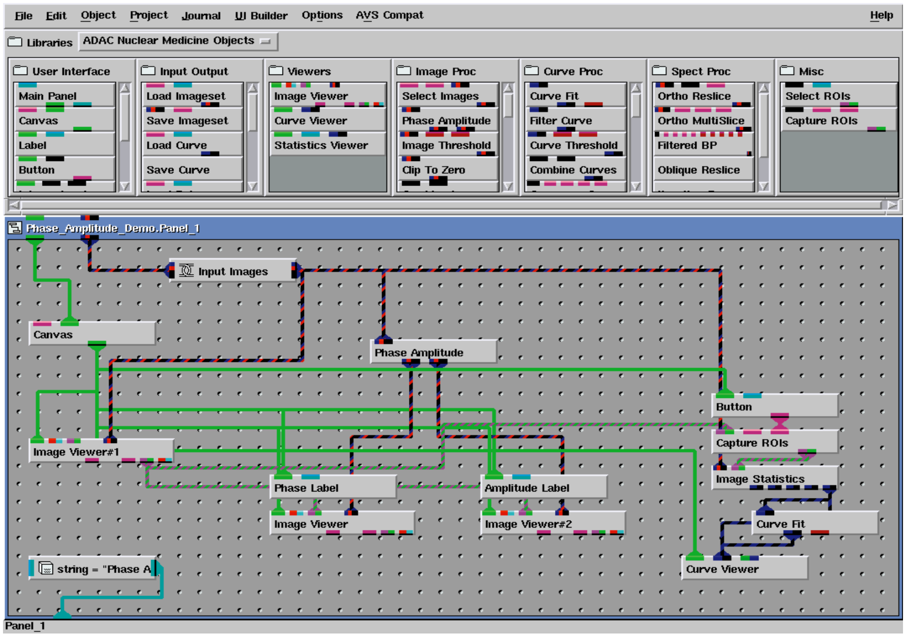


Figure 2 : Inside an application. Courtesy of ADAC Laboratories Inc.

Saving/Restoring a network in the Network Editor actually stores the application as a V file, which can be accessed through the Network Editor or through the V command-line language. Every action that you perform in the Network Editor has a corresponding V command.

5 A collection of kits

AVS/Express provides the user with hundreds of modules, that are contained within "Kits". Each Kit contains modules that share a common theme, such as data visualization (the Data Visualization Kit), graphics rendering (the Graphics Display Kit), relational database access (the Database Kit), graphing and hard copy output (the Annotation and Graphing Kit), and so on.

5.1 Data Visualization Kit

It includes a large catalog of techniques available, from 2D charts with annotation to 3D texture mapped images. There are techniques for traditional 3D representations such as colored slices through a 3D dataset or isosurfaces that show surface connecting points with the same value.The Data Visualization Kit also includes readers for the most common datasets; a menu-driven tool to create custom readers; support for NULL data (in 2D and 3D), support for Polar (including polar

[1] *Note* : In AVS/Express, unlike other packages using visual programming, linking two modules together does not connotate establishing a channel between these two modules across which data will flow. A "connection" indicates a link between the parameters of two (or more) modules. Theyshare the same memory and are notified of changes that will cause a method to execute or another parameter to be computed again.

axes), Cylindrical, and Spherical coordinates; and support for Multi-Block Data (arrays of contiguous fields of data).

5.2 Graphics Display Kit

Anything and everything related to graphics rendering is included in the Graphics Display Kit. It provides a unified viewer for geometries, images, volumes and graphs, allowing the user to produce a scene that combines all these different techniques in a single window.This scene can also be output as a Postscript file (Color, B&W, EPS), a CGM file, or as image formats like TIFF, GIF or JPEG.Rendering of displays in the Viewer is optimized :

- 2D calls are made through X11 on UNIX and GDI on PC
- 3D calls are made through OpenGL, PEXlib or XGL for best performances on supported platforms; or through X11 in the absence of hardware rendering, so that the user can perform 3D visualization even on X-terminals or low-end workstations.

The architecture used for rendering includes the following features :

- direct rendering of data without creating intermediate geometries,
- alternate object for interactive manipulation of complex objects,
- data chunking to enable rendering of large images,
- virtual palette (enables assignment of color data values in real time).

Direct user interaction within the viewer is enabled via a comprehensive user interface built into the viewer itself, but this user interface is also customizable, and additional actions can also be defined. Typical ones might include the drawing of a region of interest, to isolate an area where a computation needs to be performed, or to enable the editing of a 2D mesh simply by selecting the node to edit.For image display, tile display is supported as a standard feature.

5.3 User Interface Kit

The User Interface Kit includes all the necessary objects to create a customized user interface using a visual programming approach. It includes all the standard Motif or Windows widgets that one would expect, plus additional ones, such as dials, which can provide the user with a handy means of entering values.A user interface is constructed by simply building up the "tree structure" of the interface, and can be tested during the creation phase. As objects are connected together, they are automatically 'active' and can react to user actions.A strong feature of the AVS/Express User Interface Kit is the ease of programming callbacks; for example, popping up a file browser dialog after a button press. In AVS/Express, to add a button to a user interface that will do this, the user simply has to connect a parameter named 'do' within a "button" object, to a parameter named 'visible' within a "file dialog" object. As soon as the user presses the button in the user interface, the 'do' function gets a value of "1" (meaning execute), and so does 'visible' (because they are linked together), and the file browser dialog pops up. A user interface built on UNIX in AVS/Express can be ported to Windows by simply bringing the AVS/Express V code from the UNIX system into AVS/Express running on a PC, and then saving it out on the PC.

5.4 Image Processing Kit

The Image Processing Kit contains a number of image processing techniques, including algorithms for edge detection, contour, fast-fourier transforms, etc.

5.5 Database Kit

The Database Kit provides modules that interface AVS/Express with the most popular relational database management systems (RDBMS). Database Kit interfaces currently exist for Oracle, Sybase, and Informix. In addition, an Open Database Connectivity (ODBC)interface is provided that enables AVS/Express to interface to any relational database that supports ODBC.Multiple simultaneous database connections are allowed, so that, for instance, the user can access data stored in an Oracle database and data stored in an Informix database, within the same application.

5.6 AVS 5 Compatibility Kit

AVS/Express is a successor to AVS 5. In order to provide users with a smooth migration from AVS5 to AVS/Express, an AVS5 Compatibility Kit is provided that allows certain AVS5 networks to run within AVS/Express.

6 Building new modules

AVS/Express includes an "Add Module Tool" that provides a menu-driven interface for the creation of new modules. Users can create modules in C, C++ or Fortran. An API is available for C, C++, and Fortran to provide access to user code written in one of these languages. In the case of C++, there is a direct mapping between the V structures that are used in AVS/Express and user-defined functions in C++.The C++ interface is also designed to integrate C++ classes into AVS/Express simply by asking AVS/Express to create a 'link' between a V object and a C++ class. Then, each time the user creates an instance of the V object, AVS/Express will automatically create an instance of the corresponding C++ class. (The same things occur with the "delete" and "free" functions).These features make integrating existing code into AVS/Express extremely fast and easy.

7 AVS Developer Editor

A specific version of AVS/Express is available for developers, as described below.

7.1 Access to low-level modules

The Developer Edition gives the user access to every level of every module in all of the AVS/Express kits. (The Visualization Edition restricts access to the AVS-supplied modules to those parameters that are necessary for end-user access). In this manner, the developer can build custom macros dedicated to the particular types of data found in his particular application environment. He can also create custom interfaces for his end-users that only expose those parameters that he deems necessary.

7.2 A component generator

When a user develops modules or macros (macros are small applications that connect modules together) using AVS/Express, it is possible at any stage of development to export any and all objects (modules and macros) as standard components that can be used within other development environments.Export capabilities currently exist to generate C++ classes, or OLE/ActiveX controls (in the PC environment).

7.3 A runtime generator

To distribute an application made with AVS/Express, the developer needs to create a single executable which contains everyting related to his application. To do this with AVS/Express is as simple as pressing a button in a menu and typing in a file name. An executable is automatically created that can be distributed to others.

Since all AVS/Express Developer Edition licenses are floating licenses, the user can generate runtimes for all supported platforms by simply loading in the same application on each AVS/Express platform, and then saving out an executable.

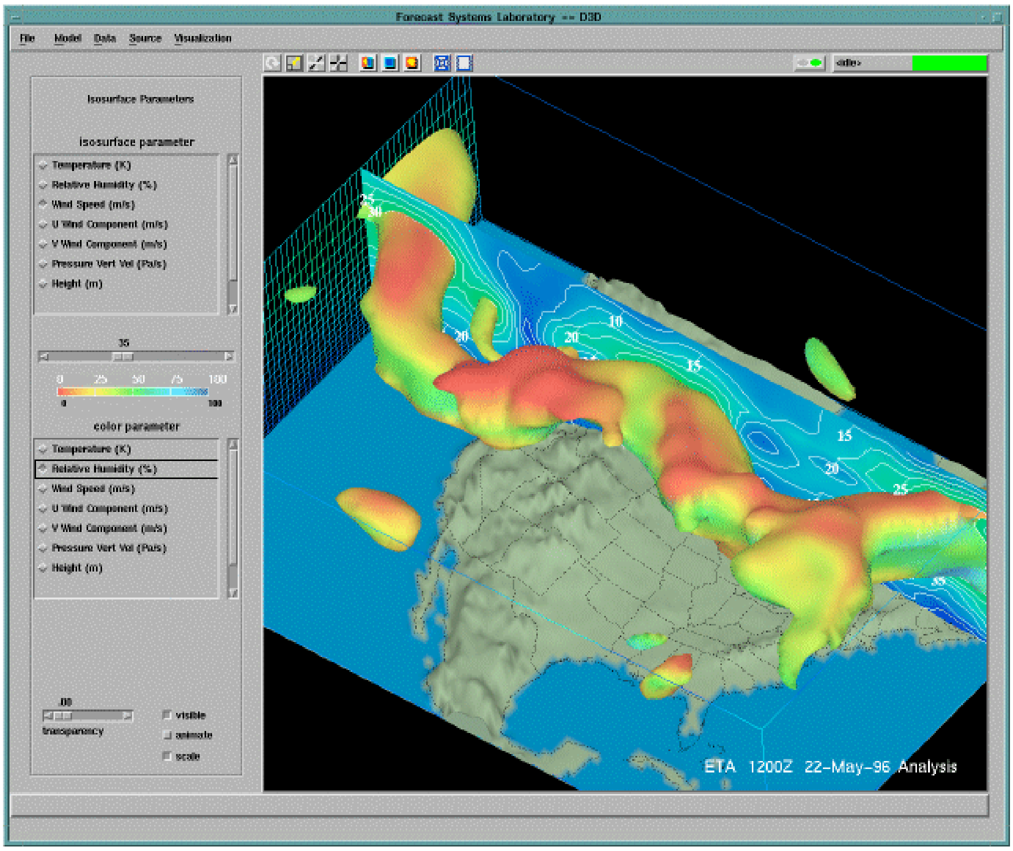


Figure 3 : AVS/Express meteorological application. Courtesy of NOAA Forecast Systems Laboratory

8 The future

AVS/Express is an evolving product and, as such, will see its feature set, which is already at a very high level, become larger and larger over time.The Internet/World Wide Web is one of the areas where AVS/Express envisions a lot of expansion in the future.Specific features related to the Web that are currently being added to AVS/Express include:

- A URL reader, to access files located anywhere on the Web,
- A VRML 2 writer to export 3D pictures to the Web and then manipulate them in 3D, using viewers that support VRML,

- The ability to create Web viewer plug-ins to provide access to sophisticated rendering techniques like 3D texture mapping or volume rendering that are currently not supported by VRML.

These and other enhancements will be coming in the next few months. But don't delay in hopping on the new train for graphics -- the Express train!