◇◇◇◇◇◇ 特集 ◇◇◇◇◇◇◇

AVS の使い方とモジュールの開発

~プログラムを書かなくてもできる高度なグラフィックス~

吉廣保1

1 はじめに

AVS(Application Visualization System) は米国 AVS 社が開発したサイエンティフィック・ビジュアラ イゼーション・システムです. プログラムを書かなくとも簡単にアプリケーションを構築できる仕組みを 提供しています. もともと,科学技術計算の結果の可視化のために開発されたソフトウエアですが,その 多機能性や柔軟さ,拡張性の高さ,使いやすさなどからアプリケーション開発のプラットホームとしても 利用されています.

AVS を使うと、2次元幾何データを対話的に表示・処理、データの2次元グラフ化、2~3次元幾何 データを対話的に表示・処理することができます。

AVS では、モジュールと呼ばれる可視化手順のうちのある機能を1つの部品にしたものを組み合わせ て、アプリケーションを構築することができます.この、モジュールの組合せを作成・編集するために ネットワークエディタが用意されています.ネットワークエディタについては後述します.また、用意さ れたモジュールで不十分な時には、新しいモジュールを製作し、登録することもできます.

センターの研究用システムに導入されたのは AVS/Express (AVS6)ですが、今回は筆者の経験豊富な AVS5 をベースに説明をしていきますので、実際に使用される時、引用している図などに多少違いが生じ るかも知れません.しかし、AVSの基本的な考え方に相違はありませんので、使用される時に違っている 部分は適宜読みかえて下さい.

2 AVS で扱えるデータ型

まず,何らかのものを可視化するには,その入力データを準備しなくてはなりません.そこで, AVS で 扱えるデータの種類について代表的なものを取り上げて説明します.

¹情報工学研究科生物化学システム工学分野

| データ型 | 内容 |
|---------|------------|
| byte | 1バイトの数字 |
| integer | 整数值 |
| float | 単精度浮動小数点数值 |
| double | 倍精度浮動小数点数值 |
| string | 文字列 |

表 1: 基本データ型

AVS では、スカラデータや文字列などのパラメータ表示に使用される基本的なデータ型と、科学技術計 算の結果などを可視化する時に用いられる、基本データ型から構成される集合データ型が存在します.(表 1,表2参照)

このように様々なデータ型が定義されていますが、もちろん、ユーザが基本データ型を用いて新たに集 合データ型を定義することができます.

入力データを準備する場合は、標準のデータ型でも、ユーザ定義のデータ型でもそのデータ型にあった フォーマットでデータファイルを作成する必要があります。詳しいデータ型のフォーマットについては、 マニュアルや京大センター広報などを参考にして下さい。

3 モジュールとネットワークエディタ

入力データファイルが出来上がったら、それを可視化するためのアプリケーションを作る必要がありま す.通常このような場合、グラフィックスライブラリを使用してプログラムを作成しますが、AVS ではビ ジュアルプログラミングと呼ばれる、マウス操作だけで簡単に可視化手順を組み立てる手法を採用してい るので、誰でも簡単にアプリケーションが作成できます.

アプリケーションの作成は、ネットワークエディタ (図1)を用いて、モジュールを結合することによっ て行なわれます.実際に作成したアプリケーションの例が図2です.

このモジュールこそ AVS の最大の特徴で,可視化のプロセスを複数に分割し,各要素をそれぞれ独立さ せ,基本単位としたものです.簡単に言えば,可視化のために必要な1つの機能のことです.これらのモ ジュールを組み合わせて,実際には図2の様にデータの流れに沿って接続することによって,アプリケー ションが構築されます.このデータの流れは,分配,結合が許されているので,多彩で複合的な表現で可 視化を行なうことができます.

また、モジュールは関数でなく、原則的にそれぞれが独立したプロセスとして提供されているため、コ

| データ型 | 内容 |
|--------------|--|
| field | 配列のインデックスでデータの並びを記述するデータ型. 空間的に均一な |
| | 格子点上にサンプル点の値がある場合に有利です.もちろん均一にならんで |
| | いなくても取り扱い可能です.これは構造化データと呼ばれます.データ |
| | ファイルの拡張子は.fld です. |
| UCD | 数値データとデータの並び方を別々に持つデータ型. 任意の格子 (セル) に |
| | よって空間が埋められていて、このセル毎にデータが存在している時に |
| | 使用されます. これは非構造化データと呼ばれます. |
| volume | field 型もしくは,後述の image 型の派生データ型です.均一な格子点上に |
| | サンプル点があり,内部のデータが byte 型に限定されています.データの |
| | サイズも 255 × 255 × 255 に制限されます.データファイルの拡張子は.dat です. |
| geom | 幾何形状を表すデータ(プリミティブ)を組合わせて複雑な形状を記述するデータ型. |
| | ポリゴンデータが基本になっています.ジオメトリビューワで使用されるデータ |
| | 型です. データファイルの拡張子は.geom です. |
| image | 色に関する整数値データの並びを持つデータ型.均一に並んだ2次元のデータ点 |
| | (ピクセル)上に,色相,彩度,明度,不透明度の値を持ちます.スキャナなど |
| | から取り込んだデータに使用されます.データファイルの拡張子は.x です. |
| PDB | Protein Data Bank のデータファイルです. X 線結晶構造解析などによって求められた |
| | タンパク質の3次元構造に関するデータ (XYZ 座標など) が収められています. |
| | ファイルの拡張子は.pdb です. |
| user defined | 上記以外でもユーザが自由にデータ型を定義できます.基本的には, Cの構造体の |
| | ような表記で表されます. |

表 2: 集合データ型

95



図 1: ネットワークエディタ (初期画面)



図 2: ネットワーク例

ンパイルの必要がなく,ビジュアルプログラミングを行ないながら即実行することが可能です.

では、モジュールについてもう少し詳しく説明します.

モジュールは、その機能によって以下の4種類に分類することができます.

- Data Input: 入力モジュール. 可視化を行なうデータをファイルなどから読み込むためのモジュール, または, カラーマップや軸データなどの生成を行なうモジュールです. 読み込み, 生成されたデータ は, ネットワークを通じて下層のモジュールへと渡されます.
- **Filters:** フィルタモジュール.上層から渡されたデータを選択したり,変換を行なうモジュールです.こ こでできたデータもまた,下層のモジュールへ渡されます.
- Mappers: マッパモジュール. 上層から渡されたデータを, AVS で表示できるデータへ変換を行なうモ ジュールです. できたデータは下層モジュールへ渡されます.
- Data Output: 出力モジュール.入力から、フィルタ、マッパを通して変換されてきたデータを受け取 り出力するモジュールです.出力先は、ディスプレイを始め、ポストスクリプトファイルや GIF ファイルなどのいろいろな形式のファイルに出力することができます.

これらのモジュールは、図1の右側上部にあるモジュールパレットにそれぞれ分類してアイコンとして配置されています.モジュールパレットはモジュールの倉庫であり、この中からユーザは必要なモジュールをマウスを使って選びだし、モジュールパレットの下にあるワークスペースへ配置し、モジュール間の接続を行なってネットワークを作成していきます.

モジュールによっては、単独で使用できるものも存在するので、モジュール1個からなるアプリケー ションも作成可能です.よって、モジュールは各種類すべて使用する必要はなく、本当に必要なプロセス だけを選んで組み立てれば良いのです.

モジュールアイコン (図 3)の上下 (入力モジュールは下のみ,出力モジュールは上のみ)には,データ の受渡しを行なうためのポートがあり,このポートはデータの型によって色分けされています.したがっ て,同じ色の出力ポートと入力ポートを接続することでデータの流れを作ることになり,ユーザは誤った 接続をすることがなく,簡単にアプリケーションを作成できます.接続されたポートどうしは,ポートの 色と同じ色の線で結ばれます.

次にネットワークエディタ上の操作について説明します.

モジュールパレットからワークスペースへのモジュールの移動は、マウスの任意のボタンでアイコンを ドラッグすることで行なうことができます。一旦ワークスペースへ移動したアイコンを移動する場合は、 マウスの左ボタンを使用します。

図 3: モジュールアイコン

ネットワーク上のデータの流れは基本的には入力から出力まで、上から下へと流れていきます.よって、入力モジュールをワークスペース上部へ配置して、その下にフィルタ、マッパモジュール、出力モジュールと配置するのが一般的な配置方法です.

必要なモジュールをワークスペースへ配置できたら、それらを接続してネットワークを作成しなくては なりません。例えば、入力モジュールの出力ポートとフィルタモジュールの入力ポートの接続ですが、出 カポートにカーソルを移動させ、マウスの中ボタンを押すと、その出力ポートから接続できるポートすべ てへ細い線が示されます。そのまま接続したいモジュールの方へマウスをドラッグすると、細い線が白色 に変わります。そこでマウスのボタンを離すとモジュール間の接続が出来上がります。

上と同様の方法で、マウスの右ボタンを使ってモジュール間の接続を切断することもできます.

必要のないモジュールをワークスペースへ引き出してしまった場合には、そのモジュールアイコンを ワークスペース右下にあるハンマーアイコン上にドラッグすることで、消去することができます.

作成したアプリケーションは、ネットワークへデータを渡してやるだけで実行が開始されます.具体的 には、入力モジュールを使用して、ファイルなどからデータを読み込むことで、そのデータが次々と処理 され、グラフィックスが出力モジュールから出力されます.

4 生物化学システム工学科の演習での使用例

前章まででは、AVSの標準的な機能について簡単に説明してきました.ここでは、情報工学部生物化 学システム工学科での3年次における計算機演習(必修)でAVSがどのように使われているかを紹介しま す.

技術分野におけるビジュアライゼーションは、データの解析のみならず、第三者に結果を理解しやすく かつ印象的に伝える有用な方法となっています.この計算機演習では、AVSを用いて実験結果などのデー タを可視化し、研究や発表のためにカラーグラフィックスを利用できるように、慣れ親しんでもらうこと を目的として行なわれています.

まず最初に、画像を扱うイメージビューワと3次元グラフィックスのモジュールを使用して、ファイル の読み込み、基本的なマウス操作、表示している画面の属性の変更などを覚えてもらいます. AVS ではマ ウス操作は統一されているので1回覚えるとあとは同じ操作を繰り返すだけでどの描画でも操作ができます.

次に,ネットワークエディタを使用して,前章で説明したネットワークの作成方法や,そのネットワー クの保存,読み込み,追加などを学んでもらいます.一通りネットワークエディタの使用方法を理解でき たところで,イメージビューワと同じ動作をするネットワークや,サンプルデータの3次元表示を行なう ネットワークを演習として組んでもらいます.

この演習の最後には、実際に良く使用される科学技術計算アプリケーションの出力結果を、効果的に3 次元表示させるところまで進みます.具体的には、分子軌道法プログラムである MOPAC を使ってペプチ ド鎖の電子状態の計算を行ない、計算結果をもとに静電ポテンシャルの3次元グラフィックスを AVS を使 用して描いてもらいます.前に示した図2は、この課題のネットワーク作成例です.

計算結果の中の静電ポテンシャルの値を簡単なプログラムを使って Field 型のデータとして作成しま す.ここで作成したデータファイルは, Read Field モジュールでネットワークに読み込まれます.ネット ワークに読み込まれた Field データは等値面を描くためのモジュールである isosurface モジュールと, 任 意の断面でポテンシャルの値を色分けして表示するためのモジュール arbitary slicer に流れていきます. 同時に, それぞれジオメトリ型データに変換されます.

また,静電ポテンシャルだけを描いても,それがどの位置のポテンシャルかがわかりませんので,タン パク質データバンク (PDB) フォーマットをもとにアミノ酸の構造を描く, pdb to geom モジュールも接 続します.このモジュールの出力は,名前の通りジオメトリ型データで出力されます.

こうして最終的に変換されたデータはすべてジオメトリ型なので、その出力には geometry viewer モ ジュールを使用します.これは、AVS のジオメトリビューワの機能をネットワークエディタで使用するた めに、モジュール化されたものです.

上の3つの出力 (等値面, アミノ酸構造, 任意の断面) は, ネットワークエディタ上で geometry viewer に入力する時に結合することができます. それぞれ1つずつでは, 絵としてはわかりづらくあまり意味を 持たないものどうしでも, それらをすべて同じ画面上に表示することで, 目で見てわかる出力に変わります.

さらにすすんだレポート課題では、構造が少しことなるアミノ酸の電子状態と静電ポテンシャル比較 するために、同じようなネットワーク2つを作成し、そのネットワーク間にデータの引き算を行なうモ ジュールを組み込みます.これにより、それぞれの構造の時の静電ポテンシャル等値面とそれらの差によ るポテンシャル等値面とを描くことができ、視覚的に比較することも可能となります.

このようなことを,演習3日のうちの2日間(4コマ)でできるようになります. それほど AVS はわかり やすく,簡単に絵を描くことができるのです.

99



arbitary slice

図 4: 演習でのデータの結合例

5 モジュールの開発

前章までで、AVSの標準的な機能と実用例について説明しました.ネットワークエディタとモジュール によるビジュアルプログラミングを行なうことで、それぞれのユーザ独自の可視化アプリを作成すること ができます.しかし、今あるモジュールの機能では満足できない時もあります.そこで最後に、AVSのも う一つの特徴である、標準の機能にユーザが簡単に独自の拡張を追加できるという点を紹介します.AVS に機能を追加、拡張するということは、モジュールを作成、変更するということです.それぞれの可視化 手順が分割されているので、途中に変更を加えることが容易になっています.

例えば、4で出てきた pdb to geom モジュールですが、演習程度のアミノ酸骨格表示であればこのモ ジュールの stick, ball and stick, ball などで表示する機能だけで十分だといえます. しかし, ball and stick モデルなどで表示して構造が理解できるのは、数十残基のペプチド鎖が限度だと思われます. しかし ながら、実際の PDB データは、何百残基という巨大なタンパク質のデータです. 既存の pdb to geom で は、どうも役不足といわざるを得ません. しかも、この pdb to geom では、 PDB データ中の原子の座標 データしか使用しておらず、その他に含まれる立体構造の情報や残基配列の情報などは一切使用されてい ません. これでは、折角のデータがもったいないです.

そこで今回は、実際の PDB データを表示する際にもっとわかりやすい表示方法を pdb to geom に追加 し、その上でただ簡単な構造を表示するだけでなく PDB データに含まれる情報をもとに物理量や、複雑な



図 5: 既存の ball and stick 表示

構造を表示する機能と, 画面中のマウス操作でその構造に関する情報を得られるような機能を追加してみ ました.

図 5,6に,既存の pdb to geom と,そのモジュールに手を加えリボンモデル表示をできるようにし,複 合表示も可能にした新しい pdb to geom の表示結果を電子伝達タンパク質シトクロム c3 を例に示しました.今回は紙面の都合上 AVS モジュールプログラミングが誰にでもできることを少しでも示すために,パ ラメータ入力部分の GUI の構築について紹介します.

AVS には標準で GUI が何種類か用意されています. (図 7) 左上から右向きに順に簡単に説明します.

- dial: 整数や浮動小数点の値を変更する際に使用します. マウスで変更する以外にも, サブウィンド ウで数値をキーボードから入力することもできます.
- slider: 整数や浮動小数点の値を変更する際に使用します.
- type-in: 文字列などをキーボードから入力する際に使用します.入力できる型は, 整数型, 実数型, 文字列型と分類できます.
- toggle button: パラメータの on/off を行ないます. ボタンが白い時が on. マウスでクリックする たびに on/off が切り替わります.
- radio button: 複数の候補の中から1つを選択するときに用います. radio button は必ずどれか1 つ選択されます.
- file browser: ファイルを読み書きする時に,ファイル名を選択,指定するために用います.



図 6: リボンモデル表示





102

/**** 原子単位の表示におけるセレクトボタン ****/

param = AVSadd_parameter("Representation","choice","Ball & Stick",
 "Ball & Stick\nBall\nStick\nColored Stick\nColored Residue\n
 Space Fill\nNone","\n");

AVSadd_parameter_prop(param,"columns","integer",2);

| Whole Type | |
|-------------------------|---------------|
| 🗭 Ball & Stick | Ball Mark No. |
| Stick | Colored Stick |
| Produced Residue | C Space Fill |
| None | |

図 8: ボタン作成プログラム例

図 9: ボタン作成例

• text browser: 文字列を表示する際に用います.

これらの GUI 部品は, AVS 関数と呼ばれる, C 言語の関数を呼び出すことで生成することができます. 例えば, 図 8のような数行のプログラムを記述するだけで, 図 9のような radio button を作ることが できます.

ここで生成されるボタンなどは、基本的にはネットワークエディタ画面の左側の細長いコントロールパ ネルと呼ばれる領域に現れます.しかし、ユーザが指定することで、新たにウィンドウを開いてその中に ボタン類を生成することもできます.

この様にして作成したボタン類も,数が増え過ぎると必要ないボタンを誤って操作することも出てくる かも知れません.そこで,コントロールパネル上やウィンドウ上に生成されたボタン類は,必要がなけれ ばモジュール中で隠すことがでるようになっています.これもまた,AVS 関数を使って行ないます.

具体的な表示例が図10です.図の真中にシトクロム c3 が表示されていて、その下にいくつか小さなウィンドウがあるのがわかると思います.それが、階層下のメニューを表示したウィンドウです.また、図の

右下にあるちょっと大きなウィンドウは、情報を表示するためのテキストブラウザです.

この様な複雑なメニューも、AVS を用いて開発すれば、 $200 \sim 300$ 行で記述することができます. 普通に、X-Window の Motif などを使って同じ GUI を記述するなら、簡単に1,000 行を越えてしまうこと でしょう.



図 10: 今回開発したモジュールの GUI

6 おわりに

AVSの機能について、ネットワークエディタとそのモジュールを中心に述べてきました. ここで述べ ました機能や使用法はほんの一例です. 普通なら、ここで締めくくりの言葉としてその他の詳しいこ とはマニュアルを参考にして下さいと記すところですが、著者らはあえて、AVSのユーザーズグルー プがフリーなモジュールを公開している International AVS CenterのFTPサイトを紹介致します (ftp://testavs.ncsc.org/). 日本国内では、大阪大学が mirror サーバとなっています. (最近はあまり更新 されてないようですが.)ここには、たくさんのフリーのAVSモジュールがソースで公開されています. おそらく必要とするモジュールはほとんど揃っているか、それに近いものが必ず存在するでしょう. その モジュールのソースはマニュアル以上の情報を与えてくれることでしょう.