第20回ビジュアリゼーションカンファレンス

アプリケーション開発ツールとしての AVS/Express

サイバネットシステム株式会社

っくる情熱を、支える情熱。 CYBERNET

本日の内容

- AVS/Express とは
- AVS/Expressを利用するメリット(アプリ開発)
- 利用例、どういう方にお勧めか
- 作成に必要なもの
- アプリケーション開発方法
- 簡単なアプリケーションの作成
- 本格的な開発を行う際のポイント
- 便利なコマンド



AVS/Expressとは、

汎用可視化ツール(ソフト、アプリケーション) 構造、流体解析等のシミュレーション結果の表示 実験、計測結果の表示



特徴:

- > 3 Dグラフィックスソフトウェア
- > モジュールプログラミング
- > 可視化機能の開発、アプリケーションの開発

特徴:モジュールプログラミング

データ読み込み、カット、コンタ、表示といった細分化された部品(モジュール) をユーザー自身が組み合わせて(プログラミング)一連の表示を行う



特徴:可視化機能の開発、アプリ開発

市販や無償配布のアプリケーションの開発が可能



MicroAVS:簡単可視化ソフト(販売)



GISDataExtractor:国土地理院データ変換ツール (MicroAVS付属ツール)

© 2014 CYBERNET SYSTEMS CO., LTD. All Rights Reserved.





Wvis: 気象情報可視化ツール(無償) 東海大学 航空操縦学専攻 新井直樹様開発 (独立行政法人 電子航法研究所)



3D AVS Player:ビューアー(無償)



AVS/Expressを利用するメリット

- 豊富な可視化モジュールやデータ読み込みモジュールの利用
- 結果を確認しながら開発可能
- 部品の組み替えで、機能追加や変更が容易
- 独自アルゴリズムの組み込みも可能
- マルチプラットフォームに対応(Windows, Linux, Mac)



AVS/Expressを利用するメリット

- 富な可視化モジュールやデータ読み込みモジュールの利用
- 結果を確認しながら開発可能
- 部品の組み替えで、機能追加や変更が容易
- 独自アルゴリズムの組み込みも可能
- マルチプラットフォームに対応(Windows, Linux, Mac)





豊富な可視化モジュール



面コンタ





流線



カット



線コンタ





等値面



ボリュームレンダリング

パーティクル





データ読み込みモジュール

FIELD型(構造格子型及び離散点を扱うAVS共通書式) Read_Field UCD型(非構造格子型を扱うAVS共通書式) Read UCD 形状データ書式(AVS共通書式) Read MGF 形状データ書式(書式は非公開) Read Geom Read Txt Columns Text形式 Read Txt Grid Read Txt Sequence Read HDF5 Field HDF5 Rd netCDF Fld netCDF Read Vis5D Vis5D Read DXF DXF Read Polygon Wavefront obj/Stanford Polygon/VTK Read Triangle RAW-tri angle/STL/SLP/TIN Read Img2Vol 複数枚の画像データをボリュームとして読み込む Read POLT3D マルチブロックデータ Read CGNS CGNS 国土地理院数値地図の1km, 250m, 50m メッシュの標高データ Read_DEM Read Gaussian Cube Gaussianデータ Read_Gaussian_Log Read PDB PDBデータ 分子動力学Lammps Read Lammps Read DICOM 医用データ Read WRF 気象モデル グリッド分析・表示システム Read GrADS ESRI社提唱のShapeファイル GISReadShape VTKファイルの読み込み Read VTK

※インターフェースについては一部制限があるものもあります。詳細はモジュールリファレンスを参照下さい。

利用例

企業や学内利用として

システム部門が社内利用のカスタムアプリケーションを開発し配布 部門内の処理を定型化し簡単なインターフェースで利用

顧客向け受託開発のベースとして

ソリューション会社が特定の企業向けのアプリケーションを開発し納品

外販ソフトとして

自社製品として開発し販売(ソルバー開発の会社)

自社製品の付属ソフトとして

計測・解析機器のハードベンダーが、結果の評価ツールとして開発しバンドル

プロトタイプとして

本格的な開発の前のテスト版として(テスト版は自作、製品版は外策)

どういう方にお勧めか

- ・配布本数が少ない
- ・顧客やユーザー毎にカスタマイズが必要
- ・配布・販売数が予想できない
- ・ユーザーのニーズを確かめながら開発したい
- ・とりあえずプロト版を開発したい
- ・提供・販売先でのカスタマイズ性も残したい

作成に必要なもの

- ・製品ライセンス: AVS/Express Developer Ediiton
- ・コンパイラ: (現行バージョン8.2の場合)
 - ▼Windows:以下の何れか

Visual Studio 2010

Visual Studio 2012

Visual Studio 2013

※VisualStudio ExpressEdition は不可

※必要に応じてIntelFotran12.1 (FORTRANでの機能追加時)

▼Linux:以下の何れか

gcc4.1

gcc4.4

gcc4.8

※gcc と同じバージョンの gfortran

▼Mac:

Apple LLVM 5.1 / gfortran4.8

※識別コードを設定しない場合は、ランタイム登録料は不要



アプリケーション作成の流れとライセンス

Step	作業	利用ライセンス	契約ライセンス
1	プロジェクト環境の作成	Viz / Dev	Viz / Devライセンス
2	(必要に応じてモジュールの追加)	Viz / Dev	Viz / Devライセンス
3	可視化ネットワークの作成	Viz / Dev	Viz / Devライセンス
4	実行ファイル(EXE)作成(コンパイル)	Dev	Devライセンス
5	動作確認	Dev	Devライセンス
6	FEATURE登録(識別コード取得)	-	Devライセンス+登録料
7	ライセンス組込み(識別コード組込み)	Dev	Devライセンス+登録料
8	動作確認(識別コード確認)	テスト用	Devライセンス+登録料
9	配布&外販	配布用	配布用ライセンス

- Viz • AVS/Express Visualization Ediiton
- Dev · · · AVS/Express Developer Edition

登録料 ・・・ランタイムライセンス登録料

ネットワークを非表示にするだけであれば、Vizライセンスでも可能

アプリケーション開発方法

2通りの作成方法

- モジュールプログラミングでの開発
 - 簡単に独自のアプリケーションを作成したい。
 - 可視化結果を見ながら機能の組み込みを行いたい。
 - マルチプラットフォーム版を開発したい。



MicroAVSの開発方法

- .NETのクラスライブラリとして開発
 - Windows専用アプリケーションを開発したい。
 - 独自のユーザーインターフェースで作成したい。

こちらの方法を確認されたい場合は、デベロッパーズ・ガイド 第2部 第16 章VC++ への組み込みを参照下さい



³D AVS Playerの開発方法

アプリケーション開発方法

2通りの方法

- モジュールプログラミングでの開発
 - 簡単に独自のアプリケーションを作成したい。
 - 可視化結果を見ながら機能の組み込みを行いたい。
 - マルチプラットフォーム版を開発したい。

本日はこちらを紹介します



MicroAVSの開発方法



3D AVS Playerの開発方法

AVS/EXPRESS

- .NETのクラスライブラリとして開発
 - Windows専用アプリケーションを開発したい。
 - 独自のユーザーインターフェースで作成したい。

こちらの方法を確認されたい場合は、デベロッパーズ・ガイド 第2部 第16 章VC++ への組み込みを参照下さい

簡単なアプリケーションの作成



AVSの構造格子データを読み込み、等値面と格子断面を表示するアプリケーション パラメータなどはAVS/Expressをビジュアルプログラミングする時と同じ。



- 新規プロジェクトを作成
 [プロジェクト >> プロジェクト新規 保存] メニューで新しいディレクト リ名を指定して作成する。
- 2. 可視化ネットワークの作成
- 3. アプリケーション保存する
- 4. アプリケーションファイルの整理 (不要な部分を削除)

AVS/Express - C:¥PROGRA~1¥AVSEXP~1						
ファイル(F) 編集(E) オブ:	ジェクト(0) プロ	コジェクト(P) ジャーナル(J) UIビルダ	ー オプショ			
🗂 Libraries Start	•	ソース・コード編集(E)				
C Weste New		コンパイル(C)	View			
		ライブラリの再読込み(R)	V 1011			
🖺 (HDF5 simple 1d) 🔺		キットの構成(C)	(Our			
🖺 (HDF5 simple 2d)	Impo	保存(S)	(Our			
		プロジェクト新規保存(A)				
님 (HDF5 simple 3d)	E Read	コンパイルしたプロジェクトを保存(P)	(Our			
国 (MD2 Glyph)	🚡 Read UCD	닐 axis 3d	B (Our			

プロジェクト下に以下のディレクトリとファイ ルが作成される。



プロジェクト作成後は、コマンドプロンプト (又はシェル)を起動し、このディクトリから Expressを起動する。

可視化ネットワークの作成

- 1. 新規プロジェクトを作成
- 可視化ネットワークの作成 作成後、データを読み込みパラメー タを操作して動きを確認する
- 3. アプリケーション保存する
- 7プリケーションファイルの整理
 (不要な部分を削除)



						A
ファイル(F) 🧍	編集(E)	オブジェクト(0)	プロシ	^ኦ ェクト(P)	ジャーナ	ール(J)
新しいアプ	リケーション	ン(N)				
アプリケーシ	アプリケーションの読込み(L)			Mappers (
アプリケーシ	アプリケーション保存(S)					
アプリケーシ	アプリケーション消去(D)			(Multithreaded)		
終了(X)	終了(X)			🚡 (adjus	t slice sp	_
[일 (GISWorldDatat] 일 (cell data mat			h	튑 (adve	ct multib	
딜 (GISDigitalElev 딜 (cell to node)				립 (adve	ct points)	
GISReadSI	ihape 🗸	퉙 (clamp)	~	립 (adve	ctor)	× [

アプリケーションファイルの整理

▼ アプリケーション保存されたもの

```
APPS.SingleWindowApp SingleWindowApp {
 MODS.Read Field Read Field < NEx=189., NEy=27.> {
   DVread field {
     Mesh Unif+Node Data Output Field:
   };
   read field ui {
     filename = "C:¥¥Program Files¥¥AVS Express¥¥data¥¥field¥¥hydrogen.fld";
     file browser {
       ok = 1;
       x = 100;
       y = 100;
     };
   };
 };
 MODS.bounds bounds < NEx=27., NEv=81.> {
   in_field => <-.Read_Field.field;
   BoundsUI {
     DVnode data labels {
       labels[];
     };
     UIradioBoxLabel {
       label cmd {
         cmd[];
       };
     };
   };
 };
 GDM.Uviewer3D Uviewer3D<NEx=189.,NEy=153.> {
   Scene {
     Top {
       child obis => {
         <-.<-.bounds.out obj,<-.<-.orthoslice.out obj,
         <-.<-.isosurface.out obj};
       Xform {
         ocenter = \{31.5, 31.5, 31.5\};
以下続く
```

▼ 実際に必要な部分



標準モジュールからの変更箇所が保存される。 変更箇所で不要な部分を削除するとコードが減るためメンテナ ンス性があがる。

横道:V言語

オブジェクト・マネージャー(AVS/Expressカーネル)とのインターフ ェースとして用いられる中間言語

登録モジュールは全て V言語で定義

モジュール接続やパラメータの定義

オブジェクト保存、アプリケーション保存で出力

C言語に似た記述

/* Windows の場合 */ #ifdef MSDOS string tmpdir<NEportLevels=1> => getenv("TEMP"); #else string tmpdir<NEportLevels=1> = "/tmp"; #endif /* グループオブジェクトの作成 */ group group<NEx=275.,NEy=44.,NEdisplayMode="open"> { int input = 0; float outpu = 23.4; string dirname<NEportLevels={2,0}> => <-.tmpdir; };

AVS/EXPRESS

注意:

一般的なプログラミング言語と違い、処理の順番は上から順に実行されるとは限らない。

© 2014 CYBERNET SYSTEMS CO., LTD. All Rights Reserved.



- 5. 登録場所に移動 Libraries を Library Workspace に変更
- 6. アプリケーションファイルの読み込み Workspace_1を選択し、[オブジェク ト >> オブジェクト読み込み] で、整 理したファイルを読み込む
- 7. キットの構成を設定する

8. コンパイル



\$	AVS/Express - C:¥work
ファイル(F) 編集(E) オブジェクト(O)	プロジェクト(P) ジャーナル(J) UIビルダー オプション(T) ヘル
名前変更(R)	
すべてのライブラリを検索(F)	3 Workspace 1 🗇 Workspace 2 🗇 Workspa
選択したライブラリを検索(I)	
ブラウザー(B)	
プロパティ(P)	
オブジェクト・エディター(0)	
モジュールの追加(A)	
ファイル読み込みモジュールの追加(M)	
オブジェクト読込み(L)	
オブジェクト保存(S)	



コンパイル

- 7. キットの構成を設定する [プロジェクト >> キットの構成]を選 択し、不要なキットのチェックを外す
- 8. コンパイル

6.で読み込んだオブジェクト(アプリ ケーション)を選択し、[プロジェク ト >> コンパイルしたプロジェクトの 保存] を行う

ディレクトリ部分に新規ディレクトリ 名を入力し、[了解] ボタンを押す

以上の操作で配布可能な実行(EXE)フ アイルが作成される。

© 2014 CYBERNET SYSTEMS CO., LTD. All Rights Reserved.







ディレクトリ名
C.¥work¥SampleApp ブラウズ
 ◎ 選択したオブジェクトの保存 ⑦ カレント・アプリケーションの保存
 ✓ オブジェクトを自動的にインスタンスする。 ● プロジェクトにネットワーク・エディターを含む。 ✓ 文を実行ファイルに組み込む。 Ø プロジェクトの実行ファイルを作成する。 Ø ソース・コード生成
了解

[※] この方法は単機能アプリの開発では有効だが、利用できるモジュール数など制限がある。



 一度整理したアプリケーションやオブジェクトファイルは、それ以降変更箇所 をテキストエディタを利用して編集する。(メニューから保存する度に、不要 な情報が書き出されるため)

モジュールの追加やパラメータ操作した際、どう記述するか分からない時は、 該当箇所を選択して、オブジェクト保存すると記述方法を確認できる。

・ 最低限必要なオブジェクトを利用(可能な範囲で)
 Read_Image を利用するよりも、実際の処理を行っているDVread_imageモジュールを利用する方が資源が節約できる



本格的な開発を行う際のポイント

最低限必要なオブジェクトを利用(可能な範囲で)





- ・オブジェクトの継承を利用する
 - AVS/Expressはオブジェクト指向で作られている。この仕組みを利用し効率的 な開発を行う。





ボタンの高さを変更するにはそれぞれ赤文字の部分を修正



ここでは分かり易いようユーザーインターフェースで記載したが、他のモジュールでも同様の記述が可能



- ・ 無理をしてV言語だけで組もうとしない
 Vで完結できれば、他のプラットフォームへの移植性は高まる。ただ、読解不明な V を記述するのであればモジュール化した方が、メンテナンス性は高まる
- ・ 部品毎に登録場所を分ける

ファイル(E) 編集(E) オブジェクト(<u>O</u>) プロジェクト(<u>P)</u> ジャーナル(<u>1</u>) <u>U</u> Iビルダー オプション(I) ヘルプ(<u>H</u>)							
🔁 Libraries Library Workspaces 🔹							
Express Functions	User Functions	 Workspace 1 Method 自作モジュー ル(メソッド: C,FORTRAN をで機能拡張 したもの)を登 録する 	 Workspace 2 UI ユーザーイン ターフェースを 登録する 	C Workspace 3 Module モジュールと ユーザーイン ターフェース をまとめたも のを登録する	 Workspace 4 App 最終的なアプ リケーションを 登録 		

本格的な開発を行う際のポイント



本格的な開発を行う際のポイント



右に行くほど大きな部品の組み合わせで構成する。 B機能のUIを変更したければMacroの「BのUI」だけを修正すれば完成系に反映される。



便利なコマンド

- 機能の切り替えに有効なコマンド
 - copy_on_change
 triggerをきっかけに特定の値を下流に流す
 - load_script
 triggerをきっかけにvファイルを読み込む
 - parse_v
 triggerをきっかけにVコマンドやVステートメントを実行する
 (C言語で言うgoto文のように使い方を間違えるとスパゲティ化するので注意)
- その他
 - Instancer

モジュールのインスタンスを制御する。UIの表示非表示に利用

exit_process

アプリケーションを終了する際に利用

- DVswitch

データの流れを制御する際に利用

※triggerとは、値が変更されたなど動作するきっかけ。



おわりに

AVS/Express のもう一つの使い方として、アプリケーション作成の基本的な流れを紹介しました。

Developer Editionをご契約の方は、本日紹介した内容を直ぐにご確認いただけます。 ご自分の作成れたネットワークのインターフェースを改良するところからお試し下さい。

アプリケーションを作成する際の手引きとして「デベロッパーズ・ガイド」があります。 アプリケーションとして配布するまでの手順や、起動画面の追加など開発を行う上で必要 な情報を紹介しています。開発をされる際は参考にして下さい。

デベロッパーズ・ガイドはオンライン・ヘルプにPDF版が組み込まれている他、保守ユー ザー様向けサイトからもダウンロードできます。

