ADAMS/Viewのカスタマイズ方法とその実例

MSC.ADAMS
Japanese User Conference 2002
Oct. 22, 2002

いすゞ自動車株式会社
CAE推進部
服部 浩二　　篠遠 誠司
程　燕青

iTiD コンサルティング
足立　泰

ISUZU

報告内容

- 目的
- ADAMS/Isuzuとは
- カスタマイズ手法
- 実例（ADAMS/Isuzu の紹介）
- まとめ

報告内容

- 目的
- ADAMS/Isuzuとは
- カスタマイズ手法
- 実例（ADAMS/Isuzu の紹介）
- まとめ

目的

技術の標準化、自動化の重要性

ADAMS/Viewではツール化が可能

ADAMSをカスタマイズし
技術蓄積のベースに

- ADAMS/Viewのカスタマイズ手法の紹介
- ADAMS/Isuzuでどのようにカスタマイズしたかを紹介

報告内容

- 目的
- **ADAMS/Isuzuとは**
- カスタマイズ手法
- 実例（ADAMS/Isuzu の紹介）
- まとめ

ADAMS/Isuzuとは

ADAMS/Isuzu
— 車両総合性能評価ベース

段階構成

- サスジオメトリ設計
- サス基礎特性などベンチテスト
- 車両走行特性評価

適応分野

- 車両操縦安定性
- 路面入力、車両強度耐久性
- 振動乗り心地

Isuzuの車両開発のニーズに応じて

ADAMS/Viewを独自にカスタマイズ

報告内容

- 目的
- ADAMS/Isuzuとは
- カスタマイズ手法
- 実例（ADAMS/Isuzu の紹介）
- まとめ

カスタマイズ手法

- ADAMS/Viewの機能
  - Command File　初期設定、基本作業など、多く使用
  - Dialog Box　GUI環境
  - Menu　メニューのカスタマイズ
  - Macro　サブルーチンとして使用
  - Function　数字や文字の簡単な処理
  - Library　Dialog Box、Macro、Functionなどの管理
- その他（ADAMS以外）の機能
  - 環境変数　マシン毎の設定、起動時読み込みコマンドの指定
  - Python　Command Fileで対処できない事項

Command File

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - 環境変数
  - Python

Command File

- 用途
  - モデルの作成
  - 初期設定
  - その他、数多く使用

→ 作業の基本

基本はCommand Windowからコピー

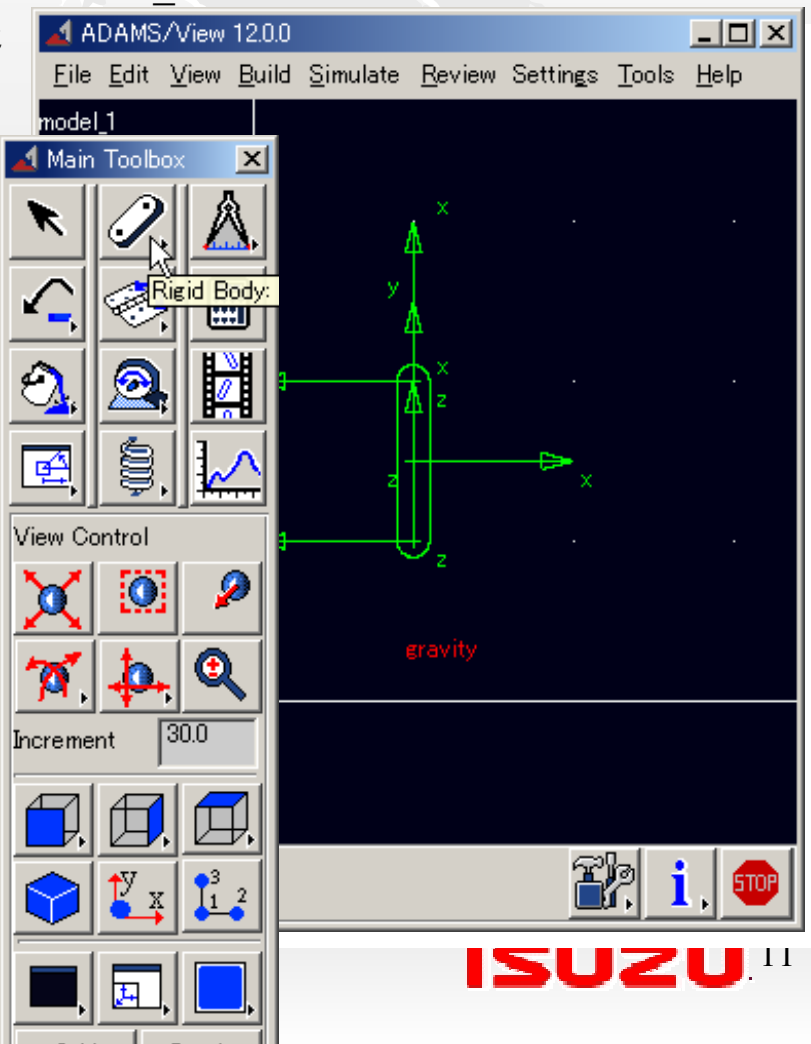
（手作業の記録）

↓

ループ、判断などを含むプログラム化

example

```
part create rigid_body name_and_position part_name=.model_1.PART_2 adams_id=2
part modify rigid_body mass_properties part_name=.model_1.PART_2
      material=.materials.steel
part attributes part_name=.model_1.PART_2 color=GREEN name_vis=off
marker create marker=.model_1.PART_2.MARKER_1 &
   adams_id=1 &
   location=50.0, 50.0, 0.0 &
   orientation=90.0, 0.0, 0.0
marker create marker=.model_1.PART_2.MARKER_2 &
   adams_id=2 &
   location=50.0, 100.0, 0.0 &
   orientation=90.0, 0.0, 0.0
geometry create shape link &
   link_name=.model_1.PART_2.LINK_1 &
   width=(10.0mm) &
   depth=(5.0mm) &
   i_marker=.model_1.PART_2.MARKER_1 &
   j_marker=.model_1.PART_2.MARKER_2
```



exampleをループ化

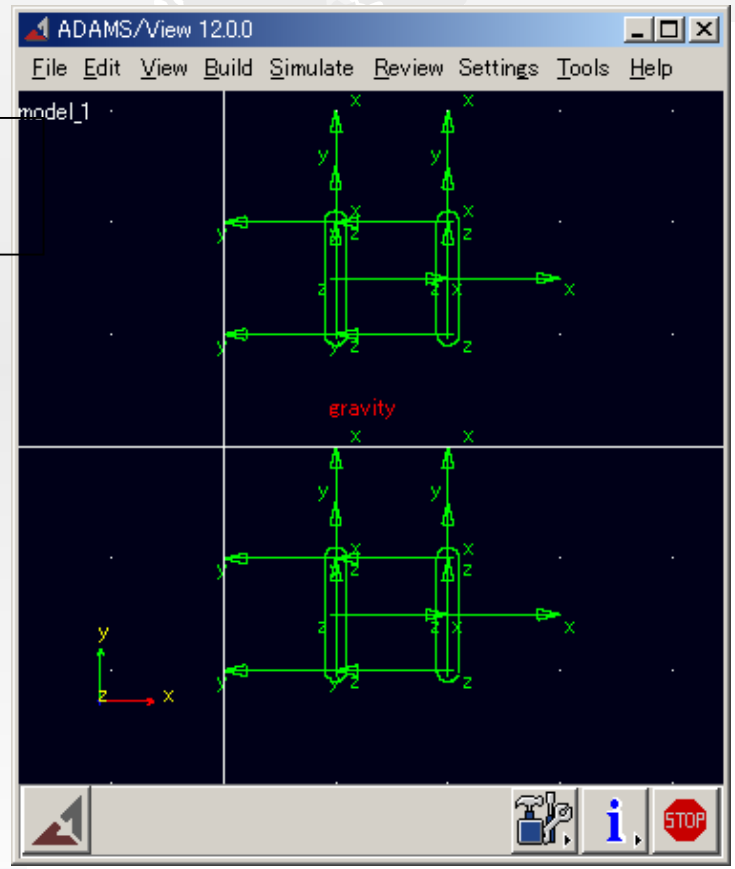
```
var set var=.model_1.FtRr str="Ft","Rr"
var set var=.model_1.RHLH str="RH","LH"
for var=.model_1.tmp1 start=1 end=2
for var=.model_1.tmp2 start=1 end=2
 var set var=.model_1.part_name &
  str= (eval(str_sprintf(".model_1.PART_%s%s",{.model_1.FtRr[.model_1.tmp1], .model_1.RHLH[.model_1.tmp2]})))
 part create rigid_body name_and_position &
  part_name=(eval(.model_1.part_name))
 part modify rigid_body mass_properties &
  part_name=(eval(.model_1.part_name)) &
  material=.materials.steel
 part attributes &
  part_name=(eval(.model_1.part_name)) &
  color=GREEN name_vis=off
 marker create &
  marker=(eval(.model_1.part_name//".MARKER_1")) &
  location=(eval(50.0*.model_1.tmp1)), (eval(50.0*(3-.model_1.tmp2*2))), 0.0 &
  orientation=90.0, 0.0, 0.0
 marker create &
  marker= (eval(.model_1.part_name//".MARKER_2")) &
  location=(eval(50.0*.model_1.tmp1)), (eval(100.0*(3-.model_1.tmp2*2))), 0.0 &
  orientation=90.0, 0.0, 0.0
 geometry create shape link &
  link_name= (eval(.model_1.part_name//".LINK_1")) &
  width=(10.0mm) &
  depth=(5.0mm) &
  i_marker= (eval(.model_1.part_name//".MARKER_1")) &
  j_marker= (eval(.model_1.part_name//".MARKER_2"))
end
end
```

ループ

Part Name

赤字の部分
（パート名と座標）を修正

座標計算



Command File

- その他の用途
  - 単位系や重力の設定
  - マクロ、ファンクションなどの作成
  - メニューの設定

多くの場所で使用

Dialog Boxやマクロのベースとなる

Dialog Box

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - 環境変数
  - Python

Dialog Box

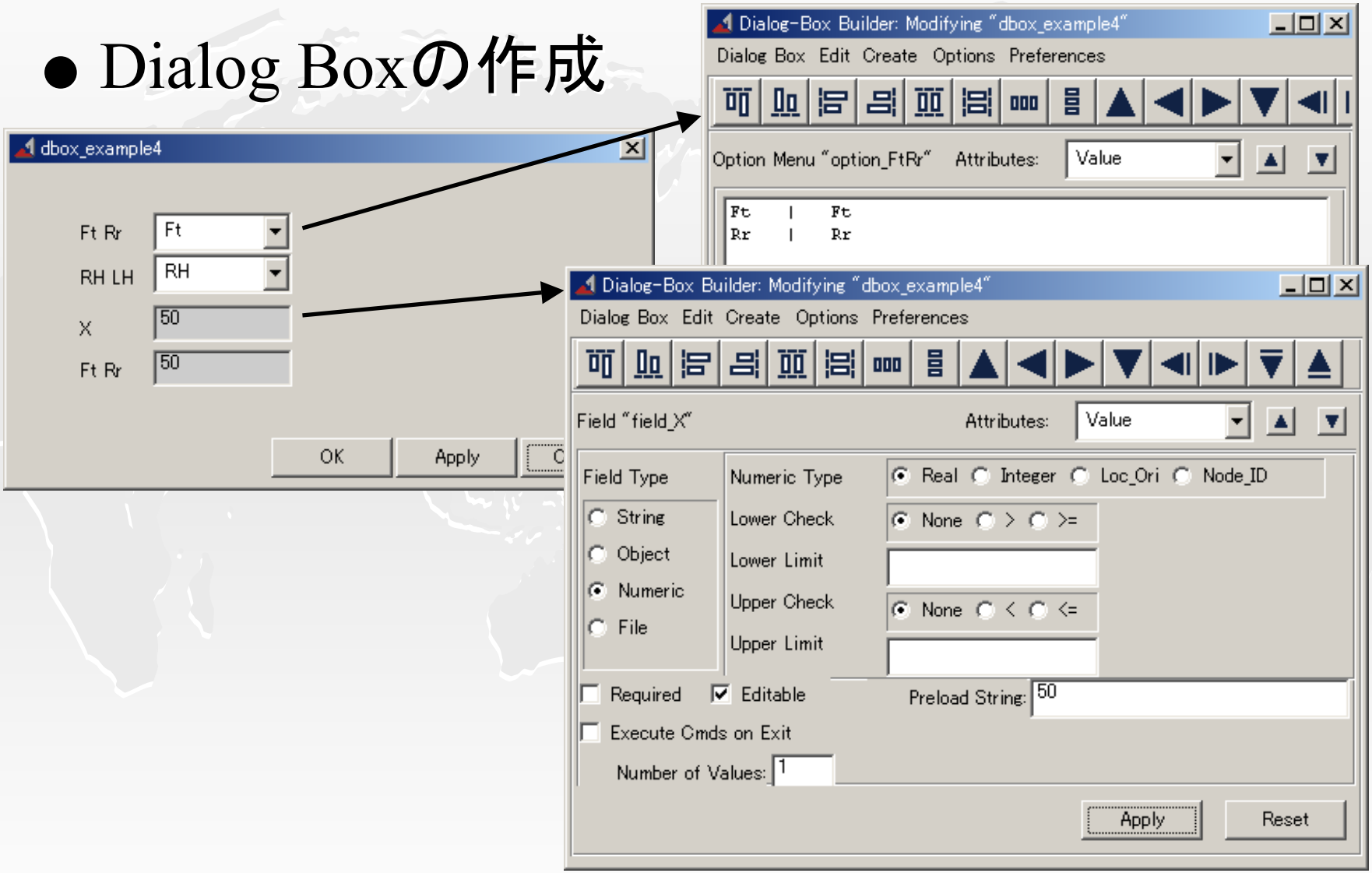
- 用途
  - GUI環境の構築
    - モデル作成
    - 計算条件設定、実行
    - パラメータの設定ができる

ユーザーの作業を
判りやすく

**Dialog Box内のパラメータを利用した
commandの実行**

example

- Dialog Boxの作成

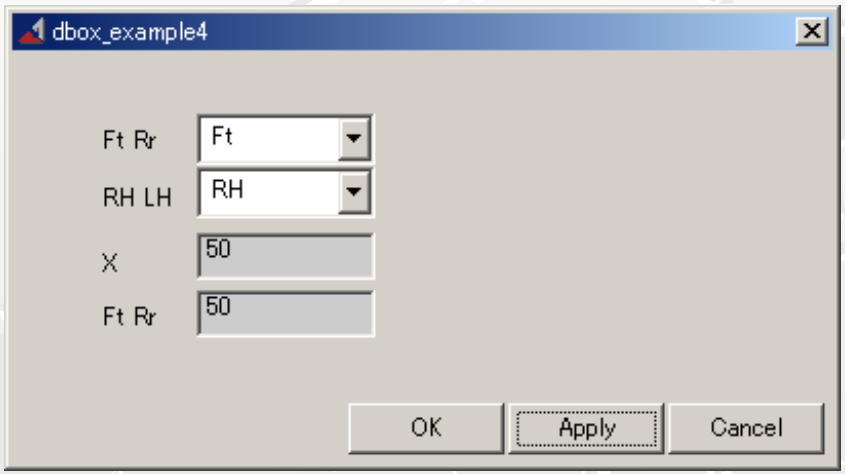


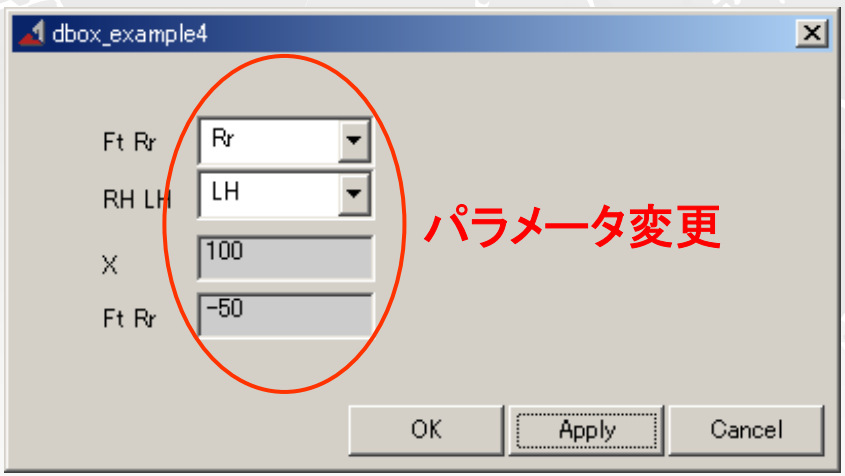
● Dialog Boxのcommandの設定

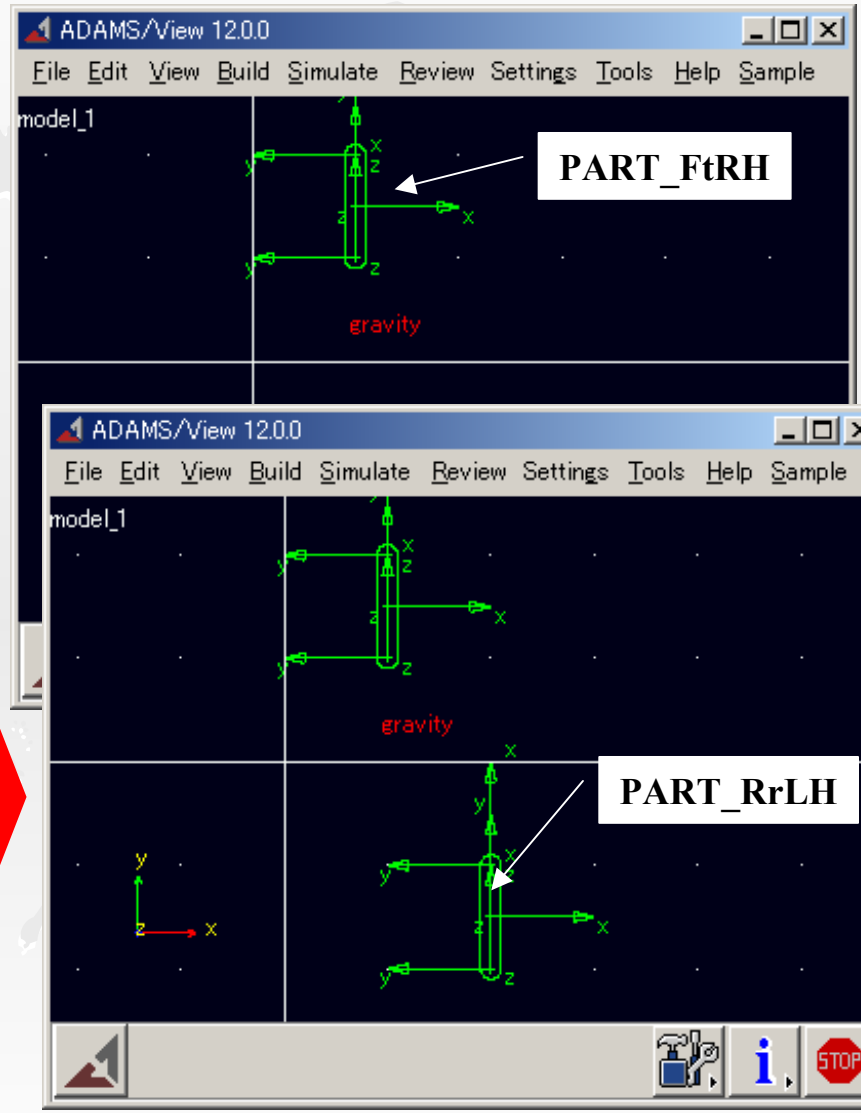
赤字の部分
（パート名と座標）を修正

```
part create rigid_body name_and_position &
    part_name=.model_1.PART_$'option_FtRr'$'option_RHLH'
part modify rigid_body mass_properties &
    part_name=.model_1.PART_$'option_FtRr'$'option_RHLH' material=.materials.steel
part attributes part_name=.model_1.PART_$'option_FtRr'$'option_RHLH' &
    color=GREEN name_vis=off
marker create marker=.model_1.PART_$'option_FtRr'$'option_RHLH'.MARKER_1 &
    location=$field_X, $field_Y, 0.0 &
    orientation=90.0, 0.0, 0.0
marker create marker=.model_1.PART_$'option_FtRr'$'option_RHLH'.MARKER_2 &
    location=$field_X, ($field_Y*2), 0.0 &
    orientation=90.0, 0.0, 0.0
geometry create shape link &
    link_name=.model_1.PART_$'option_FtRr'$'option_RHLH'.LINK_1 &
    width=(10.0mm) &
    depth=(5.0mm) &
    i_marker=.model_1.PART_$'option_FtRr'$'option_RHLH'.MARKER_1 &
    j_marker=.model_1.PART_$'option_FtRr'$'option_RHLH'.MARKER_2
```

Example実行


dbox_example4
Ft Rr
Ft
RH LH
RH
X
50
Ft Rr
50
OK
Apply
Cancel


dbox_example4
Ft Rr
Rr
RH LH
LH
X
100
Ft Rr
-50
パラメータ変更
OK
Apply
Cancel


ADAMS/View 12.0.0
File Edit View Build Simulate Review Settings Tools Help Sample
model_1
PART_FtRH
gravity
ADAMS/View 12.0.0
File Edit View Build Simulate Review Settings Tools Help Sample
model_1
gravity
PART_RrLH
STOP

Menu

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - 環境変数
  - Python

Menu

- 用途
  - メニューのカスタマイズ
    - オリジナルのツールを呼び出せるようにする
    - 解析者とユーザーでメニューを変える

ユーザーの作業を
判りやすく

Macro

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - 環境変数
  - Python

Macro

- 用途
  - パラメータ化したい作業
  - 繰り返し行われる作業
  - 他のツールから呼び出したい基本的な作業

サブルーチン的な使い方
オブジェクト指向

Command、Dialog Boxより
作りやすく汎用性が高い

example

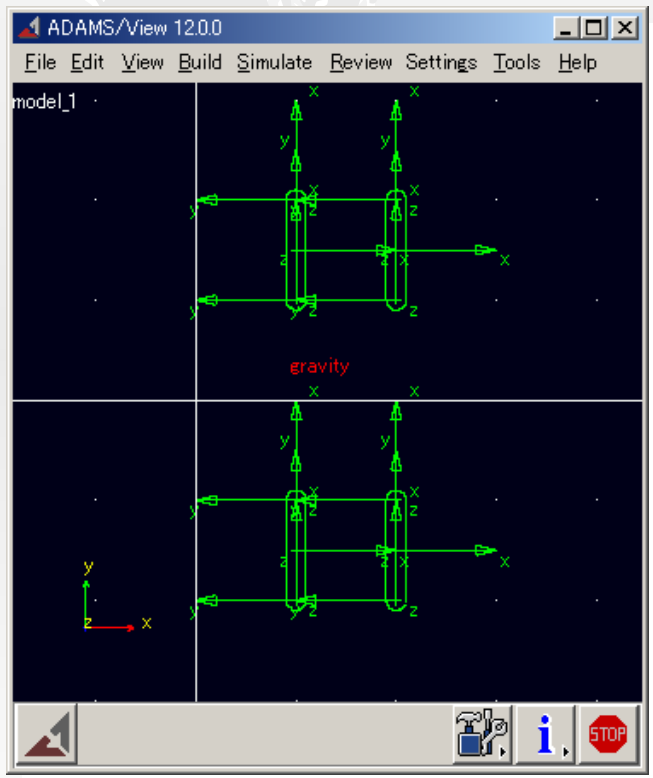
```
!$FtRr:T=list(Ft,Rr):D=Ft
!$RHLH:T=list(RH,LH):D=RH
!$X:T=real:D=50
!$Y:T=real:D=50

part create rigid_body name_and_position &
    part_name=.model_1.PART_$'FtRr'$'RHLH'
part modify rigid_body mass_properties &
    part_name=.model_1.PART_$'FtRr'$'RHLH' material=.materials.steel
part attributes part_name=.model_1.PART_$'FtRr'$'RHLH' &
    color=GREEN name_vis=off
marker create marker=.model_1.PART_$'FtRr'$'RHLH'.MARKER_1 &
    location=$X, $Y, 0.0 &
    orientation=90.0, 0.0, 0.0
marker create marker=.model_1.PART_$'FtRr'$'RHLH'.MARKER_2 &
    location=$X, ($Y*2), 0.0 &
    orientation=90.0, 0.0, 0.0
geometry create shape link &
    link_name=.model_1.PART_$'FtRr'$'RHLH'.LINK_1 &
    width=(10.0mm) &
    depth=(5.0mm) &
    i_marker=.model_1.PART_$'FtRr'$'RHLH'.MARKER_1 &
    j_marker=.model_1.PART_$'FtRr'$'RHLH'.MARKER_2
```

赤字の部分
（パート名と座標）を修正

Macro

- 繰り返し作業
  - exampleのマクロを次のcommandで実行

```
macro_sample FtRr=Ft RHLH=RH x= 50 y= 50
macro_sample FtRr=Ft RHLH=LH x= 50 y=-50
macro_sample FtRr=Rr RHLH=RH x=100 y= 50
macro_sample FtRr=Rr RHLH=LH x=100 y=-50
```



このようなMacroやCommandを使い、
モデルの組み立てや計算条件の設定を行う

先ほどのDialog Boxの例もこのマクロを
呼ぶだけにする方が判りやすい

Function

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - 環境変数
  - Python

Function

- 用途
  - 数字や文字の簡単な処理

Functionの組み合わせでつくるので、
できることに限界がある

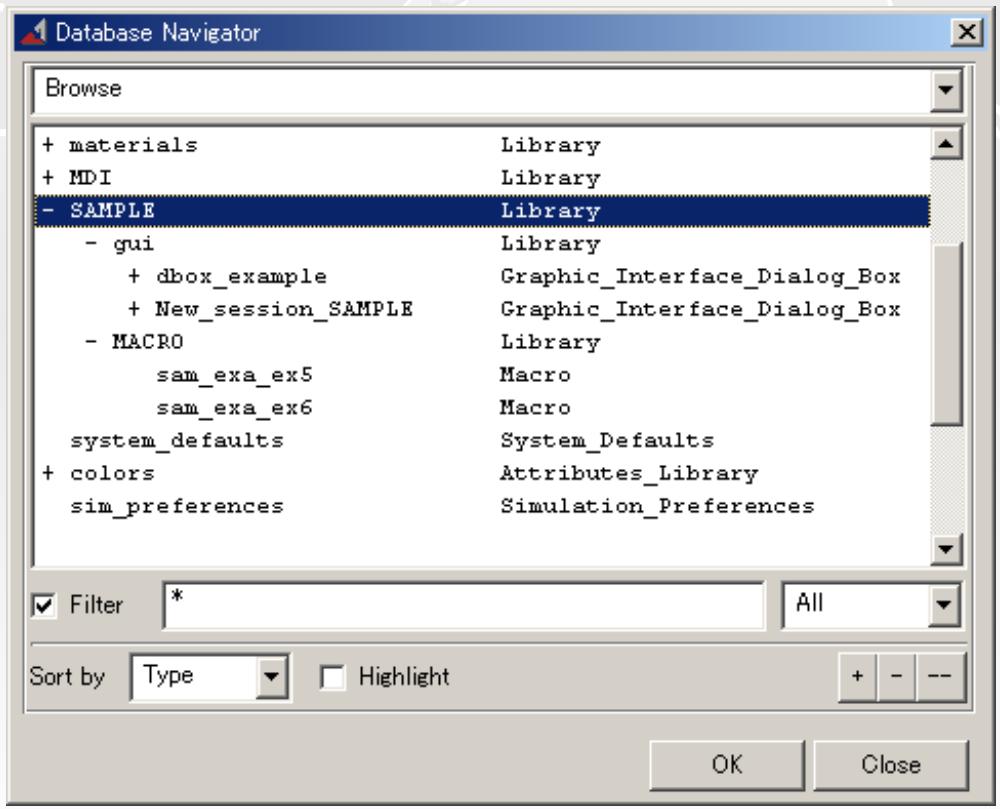
しかし、いくつかのFunctionは
なくてはならないものになっている

Library

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - 環境変数
  - Python

Library

- 用途
  - Dialog Box、Macro、Functionなどの管理



環境変数

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - **環境変数**
  - Python

環境変数

- ディレクトリの指定
  - ADAMS/Isuzuのディレクトリ
  - ADAMS/Viewのインストールディレクトリ

    ⇒ マシン毎の差を吸収
- ユーザータイプの指定
  - 解析者かユーザーか

    ⇒ GUIを変更

ADAMSのgetenvファンクションで取得し活用

環境変数

- 環境変数　MDI_AVIEW_SRCH_PATH
  - ファイル選択時、マウス右クリックで表示されるディレクトリの設定するファイルを指定
  - ADAMS/View起動時読み込みコマンドファイル（AViewAS.cmd）の場所指定

これらの設定により、ADAMSインストールディレクトリ内の変更をせず起動時実行ファイルの指定が可能になる

Setup Guides
Running ADAMS on Windows
Setting Preferences for ADAMS
Setting the ADAMS Search Paths

Python

- ADAMS/Viewの機能
  - Command File
  - Dialog Box
  - Menu
  - Macro
  - Function
  - Library
- その他
  - 環境変数
  - Python

Python

- Command Fileを作成して実行することが多い
  - Command Fileの文法はプログラム言語としては弱い
  - ファイルI/Oなどの機能が少ない
  - ループや配列などの処理が遅い
  - 以前はcshell上でawk、perlなどを用いていたが、移植性が良くない

Python

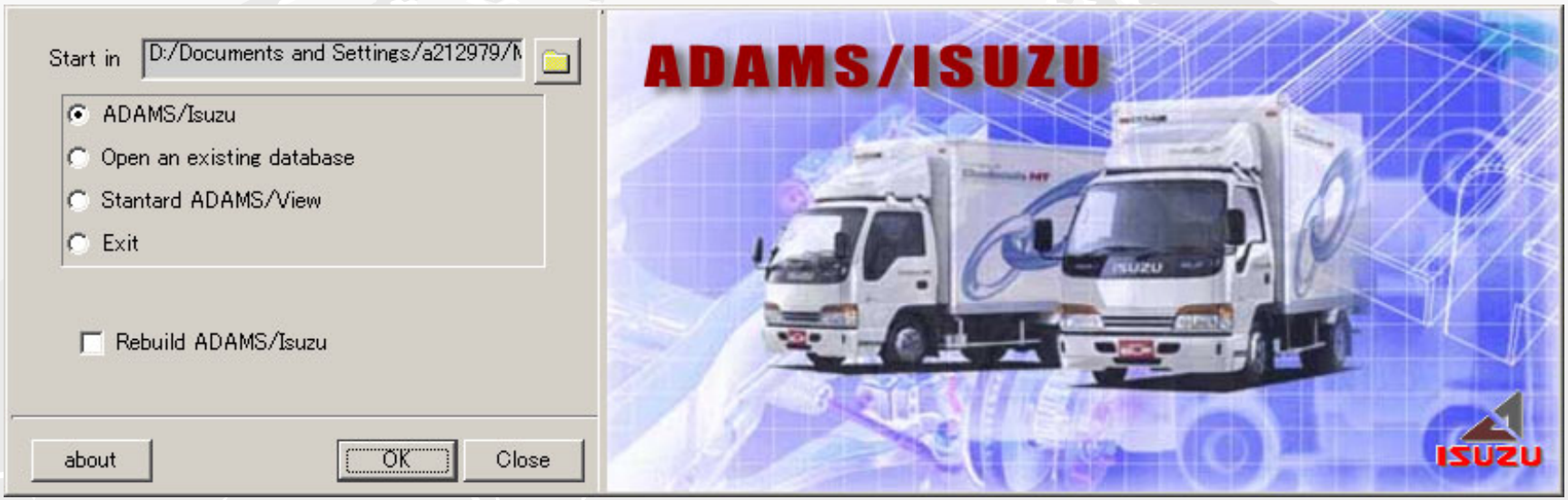
Python

- Pythonとは
  - スクリプト言語の一種
  - パブリックドメイン
- Pythonはどのマシンでも使える
  - ADAMSインストールディレクトリ内にある
  - mdi pythonでPythonインタープリタが立ち上がる
- 用途
  - MacroなどからSystemコマンドでPythonを実行し、Command Fileを作成し読み込む
  - 主にファイルの読み書き、重たい作業の高速化を目的として使っている

報告内容

- 目的
- ADAMS/Isuzuとは
- カスタマイズ手法
- 実例（ADAMS/Isuzu の紹介）
- まとめ

ADAMS/Isuzu（起動）

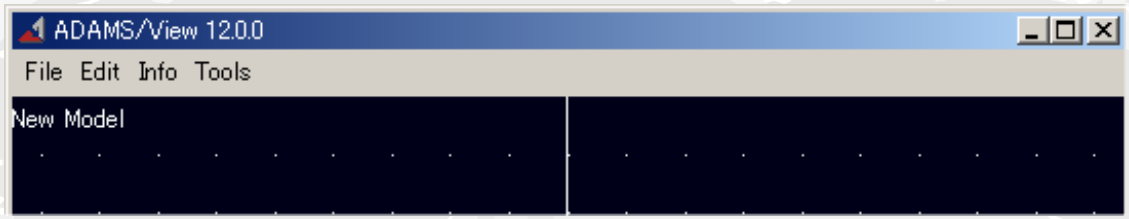


起動時表示Dialog Box

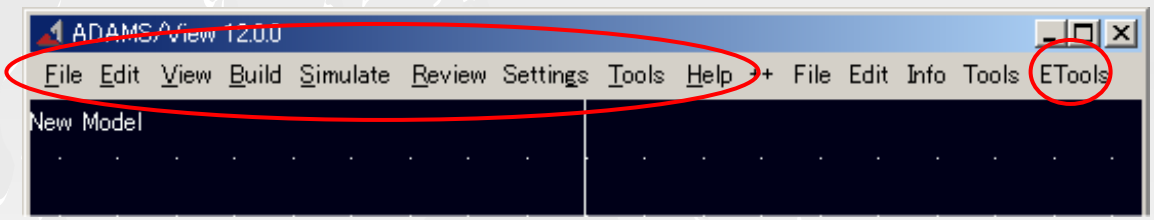
- ADAMSインストールディレクトリ内は無変更
  - ADAMS version UP時もすぐに対応
- ADAMS/Isuzuインストールディレクトリはどこでも可能
  - UnixとPCでも共有可能

ADAMS/Isuzu（Menu）

- ユーザー用メニュー



- 解析者用メニュー



- 起動後に変更が可能
- 環境変数でデフォルト設定が可能

ADAMS/Isuzu（Macroの例）

- 初速の設定
  - 全てのパートの初速度を設定
  - 回転体は回転速度も設定する
- ファイルのコピー
  - Systemコマンドを用いてファイルをコピー
  - OSを判断してコマンドを変えている

    ⇒Systemコマンドは全てMacroから呼ぶ

ADAMS/Isuzu（Macroの例）

- テキストファイルのロード
  - テキストファイルの内容を指定した変数に読み込むマクロ

Test.txtの内容

```
MSC.ADAMS
Japanese User Conference 2002
Oct. 22, 2002
```

File="Test.txt" Variable=.model_1.test
を引数としてマクロ実行

ファイルの内容を
変数の値に

.model_1.test を作成（変更）

```
Object Name      :  .model_1.test
Object Type      :  Variable
Parent Type      :  Model
String Value(s)  :  MSC.ADAMS
                    Japanese User Conference 2002
                    Oct. 22, 2002
```

Pythonを利用して作成

ADAMS/Isuzu（Macroの例）

- テーブルデータの読み込み

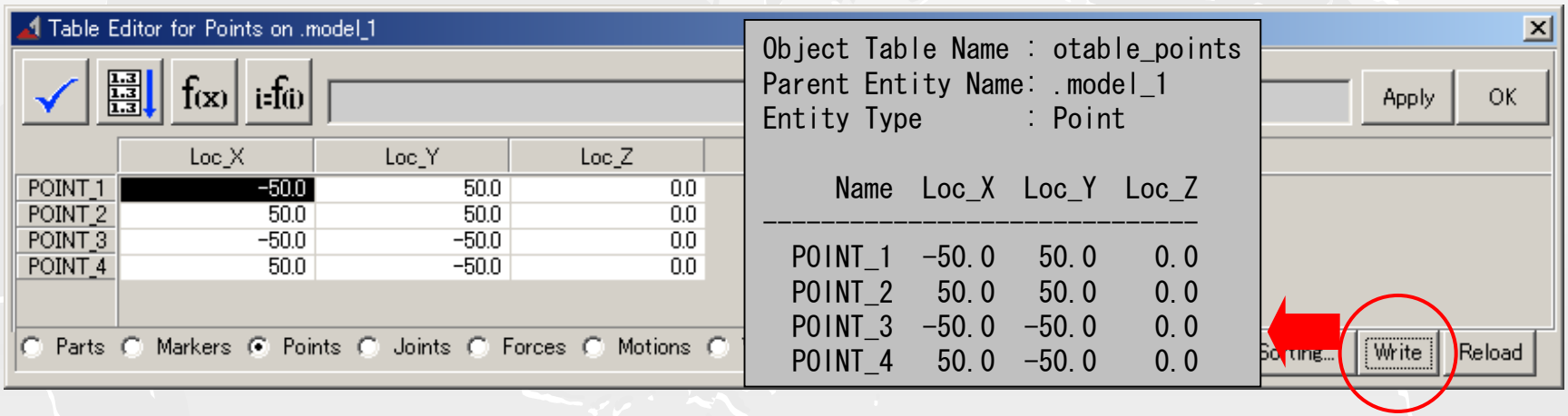


Table EditorのWriteボタンでセーブしたデータを読み込むマクロ

Pythonを利用して作成

ADAMS/Isuzu（Functionの例）

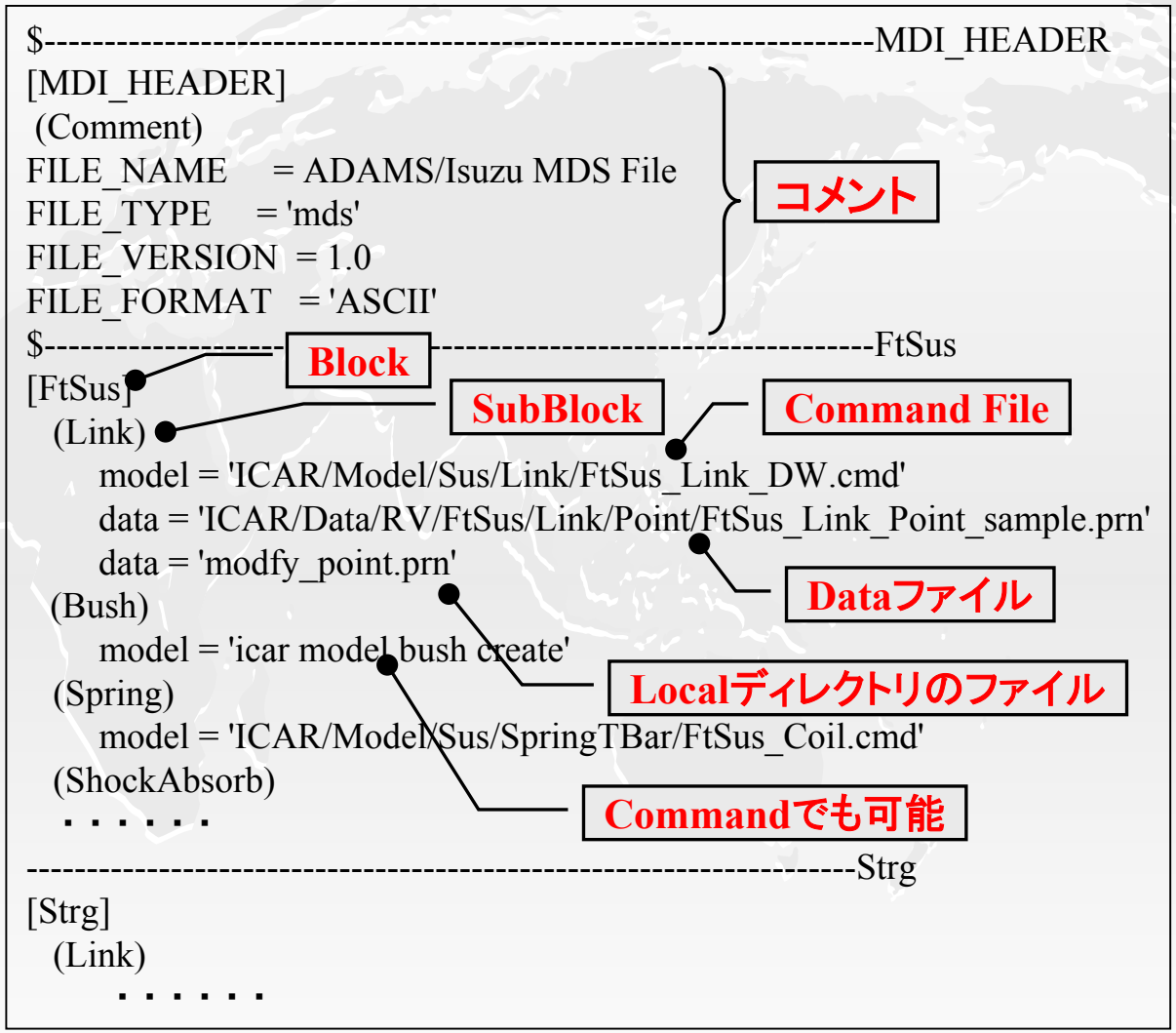
- 座標関連
  - {x,y,z} を与えると {x,-y,z} を返す
  - 2組の座標を与えると中点の座標を返す
- 文字列操作関連
  - "¥"を"/"に変換　日本語Windowsのパス名用
    - ADAMS/Isuzu内は"/"で統一
  - 文字列置換
    - str_xlateが標準で用意されているが、文字数が異なると変換できない
    - str_xlate2("abc_123","abc","A") ⇒ "A_123"

ADAMS/Isuzu（Functionの例）

- ファイル名関連
  - パス名を返す
  - パスの無い名前を返す
  - 拡張子をなくした名前を返す
  - Function FC
    - 先頭が「ICAR/」かで判断し
      - FC("ICAR/file1") ⇒ "c:/local/ICAR/file1" 変換
      - FC("other/file1") ⇒ "other/file1" そのまま
      - FC("c:/other/file1") ⇒ "c:/other/file1" そのまま
    - ディレクトリの指定（c:/local/）は環境変数で行う

      ⇒ディレクトリの移動に対応

ADAMS/Isuzu（モデルファイル）



ファイルの内容をVariableに読み込み

Block,SubBlock単位で実行するかしないか設定可能

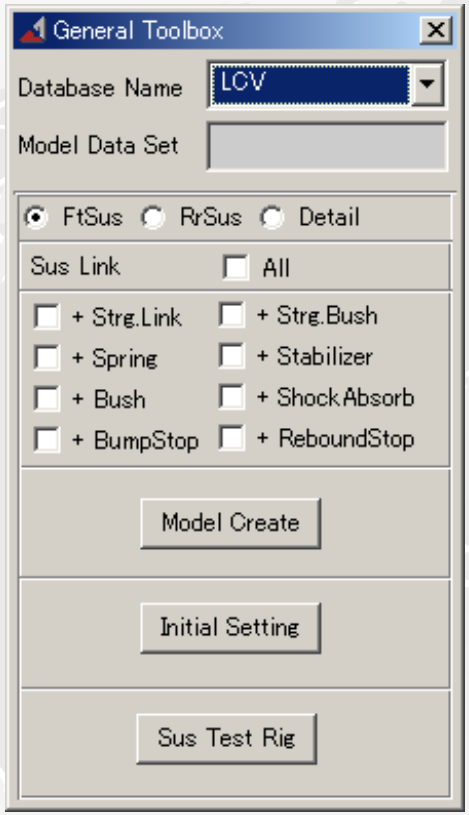
Model=はCommand FileとCommandを設定可能

Dataファイルはテーブルエディタでセーブしたファイル

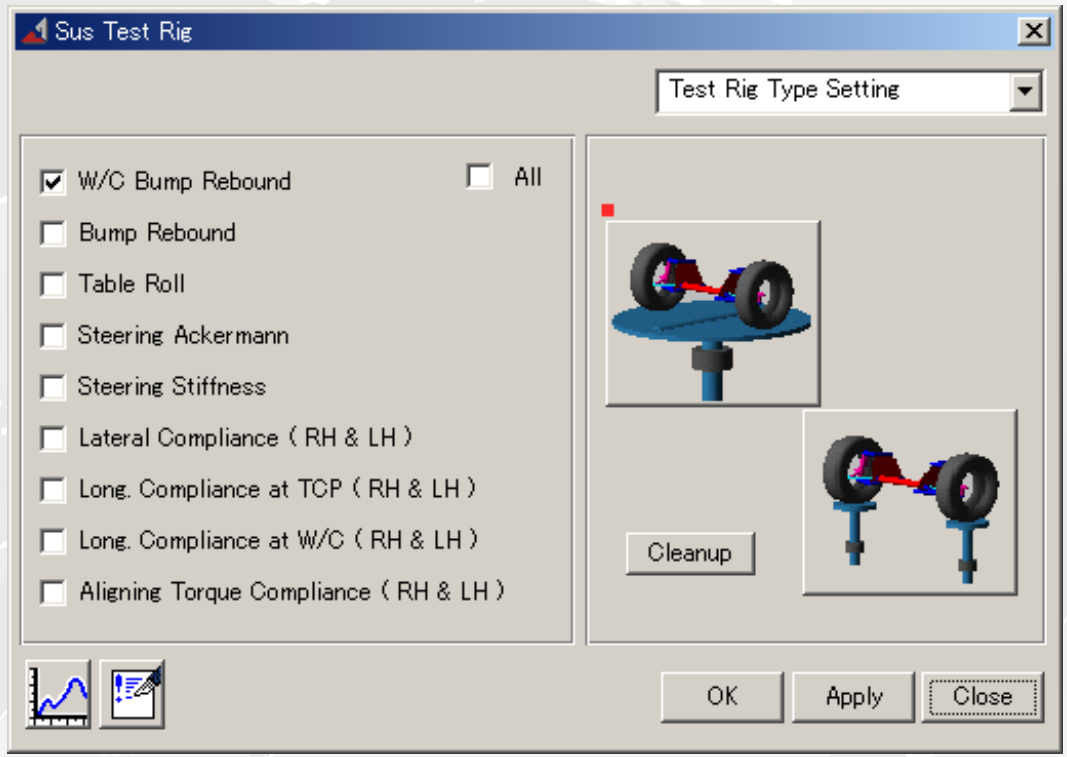
ファイル名はICAR/から書くそれ以外だとLocal

MacroやFunctionでこのような機能を実現

ADAMS/Isuzu（Dialog Boxの例）



モデル作成の例



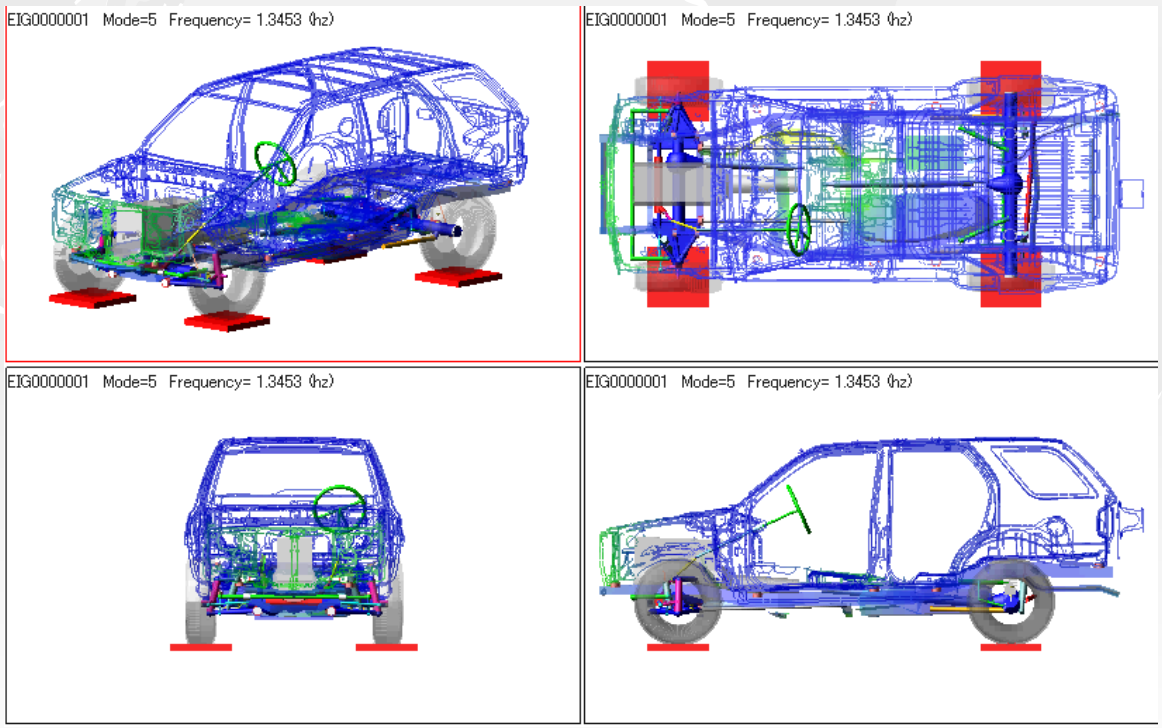
計算ツールの例

ADAMS/Isuzu（ツールの例）

- 固有値解析結果処理
  - 固有周波数一覧ホームページ作成
  - リンク先のaviファイル作成

Result of Eigen Value Analisys

mode	aviへリンク	aspへリンク
001	1.3Hz	1.3Hz
002	1.4Hz	1.4Hz
003	2.6Hz	2.6Hz
004	2.9Hz	2.9Hz
005	3.1Hz	3.1Hz
006	5.4Hz	5.4Hz



報告内容

- 目的
- ADAMS/Isuzuとは
- カスタマイズ手法
- 実例（ADAMS/Isuzu の紹介）
- まとめ

まとめ

- ADAMS/Viewカスタマイズ手法
  - Command File、Dialog Box、Macroなど、
    基本的なカスタマイズ手法を紹介した
- 実例
  - ADAMS/Isuzuでこれらの手法をどのように活用し
    使っているかを紹介した
- ADAMS/Viewカスタマイズの勧め
  - 確立された技術を一般化でき、新たな技術開発につながり、
    技術を他のユーザーも使える環境ができる
    ⇒技術の標準化・技術蓄積
  - 解析者がカスタマイズ手法を知っていれば、
    ニーズに応じたモデルや検討手法を早く作ることができる
    ⇒技術開発の効率化

ADAMSを使う上でなくてはならないものに

要望

- 情報の提供
  - カスタマイズ方法
    - コマンドの機能、使い方
- 機能（コマンドやファンクションなど）の追加
  - ユーザーレベルでも対応でき、一般的に使えそうな機能はADAMS/View標準で備えて欲しい
    - ファイル（テキスト、テーブルデータ）読み込みコマンド
    - 文字列、ファイル名関連のファンクション
  - ユーザーから一般的な機能の要求
    - スプラインデータの編集（Curve Manager）
    - プロットフォーマットのセーブ