gt4f90io: gtool4 規約に基づく Fortran90 netCDF I/O ライブラリ

北海道大学 理学院 宇宙理学専攻
地球流体力学研究室 D2
森川 靖大

目次

- (大) 目標
- はじめに
- gtool4 プロジェクト (1999〜2002)
- gtool4 プロジェクト (2003〜2006)
- gt4f90io
- プロジェクト進捗状況
- まとめ

(大) 目 標

惑星大気大循環モデルの開発

○惑星大気の条件設定を手軽に変更

※大気の成分, 水の有無, ダストの有無, etc…

○こんな実験できるといいな

※GCM による地球と火星の比較惑星実験

http://photojournal.jpl.nasa.gov/catalog/

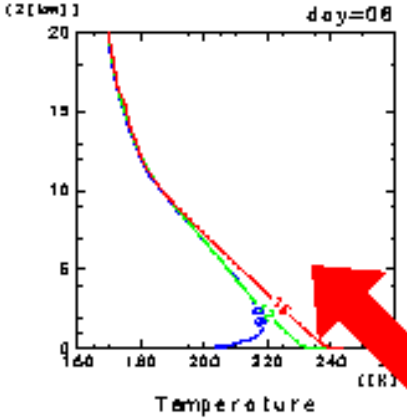
http://earthobservatory.nasa.gov/Study/LivingEarth/

大気大循環モデルでの研究では

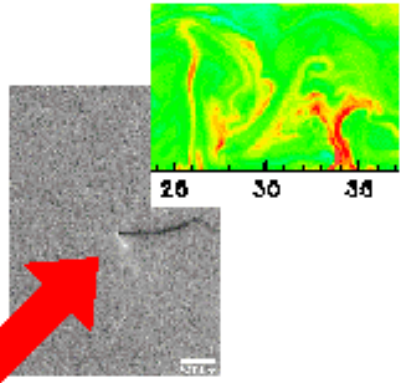
❍1D モデル, 2D モデルとの比較

❖3D モデルの結果を理解するのに必要

1D



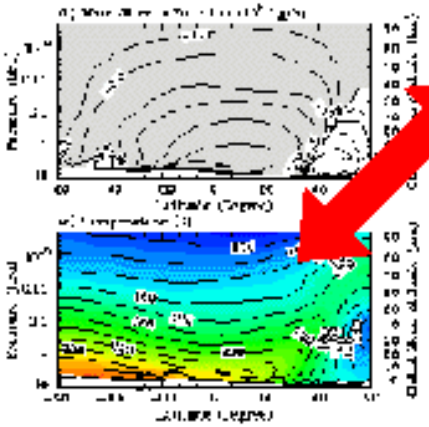
放射対流モデルによる
鉛直大気構造の理解



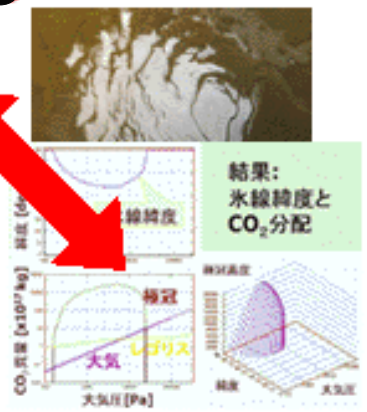
雲解像モデルによる
ダストデビルの再現

2D

3D



GCM による大気大循環の研究



EBM による極冠の
成長・後退の計算

モデル，データ共有の必要性と問題点

❍必要性

- ❖一人で 1D ～ 3D のモデルを作るのは大変
- ❖それぞれのモデルは別々の研究者が作ることに
- ❖研究者仲間でモデルやデータをやりとり

❍問題点

- ❖他人のモデルは分からない
- ❖他人のデータも分からない

どうすれば簡単に共有できるか？

- ソースコードの書き方の工夫
  - 例: 気象庁コーディングルール, AGCM5, FMS マニュアル, GFD Dennou Club dcmodel プログラミングガイドライン
- プログラムの構造の工夫
  - 例: FMS, DCPAM (spml, gt4f90io, ISPACK)
- ドキュメントの整備の工夫
  - 例: RDoc Fortran 90/95 解析機能強化版
- モデルが入出力するデータに関する工夫

(改めまして)

はじめに

データの嵐

- 様々な形式のデータが多量に氾濫
  - 数値モデルデータの種類，数量の増加
    - モデルの実装，当面の要望に従って様々な種類が
    - 計算機の性能の向上により，数多くのデータが
  - 観測データの種類，数量の増加
    - 観測機器の実装，当面の要望によって様々な種類が
    - 観測機器の性能の向上により数多くのデータが

データ参照コストの増加

❍データ参照コスト

- ❖ そのデータが何者であるかを知るための手間暇のこと
  - 「何」のデータ？「いつ」,「どこで」,「誰が」作ったデータ？どうやって「見る (可視化)」？ どうやって「いじる (解析)」？ … etc

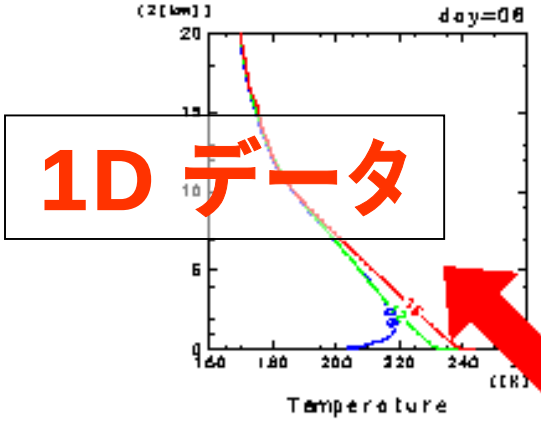
❍種類の増加に伴うコストの増加

- ❖ 例: 人からもらったデータ, Web からダウンロードしたデータを扱うための手法の調査だけで結構手間.

❍数量の増加に伴うコストの増加

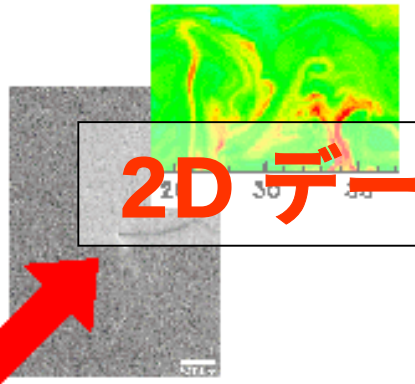
- ❖ ちゃんとデータフォーマットを決めているのでない限り, どんどん増えるデータはその時々に応じて質的にも変化する.
  - 例: 先月, 先週に作ったモデルのデータ, すぐに使えますか??

階層的モデル群による惑星大気の研究でも..



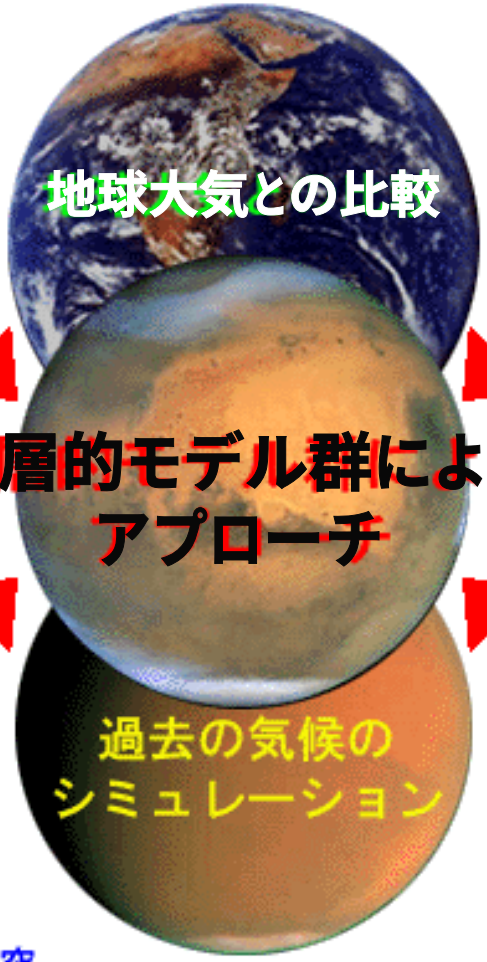
1D データ

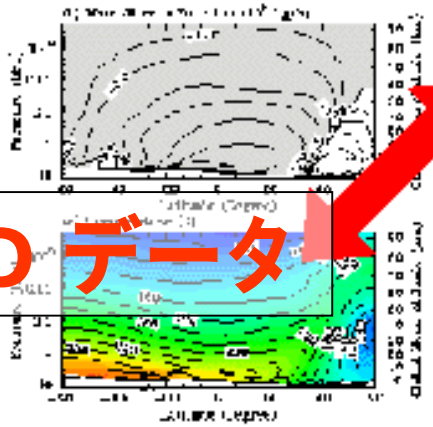
放射対流モデルによる
鉛直大気構造の理解



2D データ

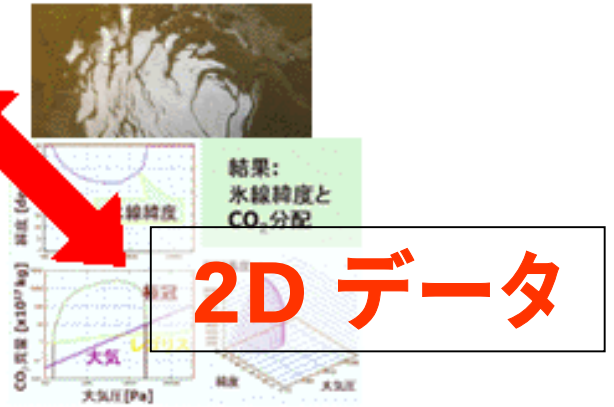
雲解像モデルによる
ダストデビルの再現





3D データ

GCM による大気大循環の研究



2D データ

EBM による極冠の
成長・後退の計算

❍効率的なデータ相互参照が必要

データ参照を効率的に行うためには？

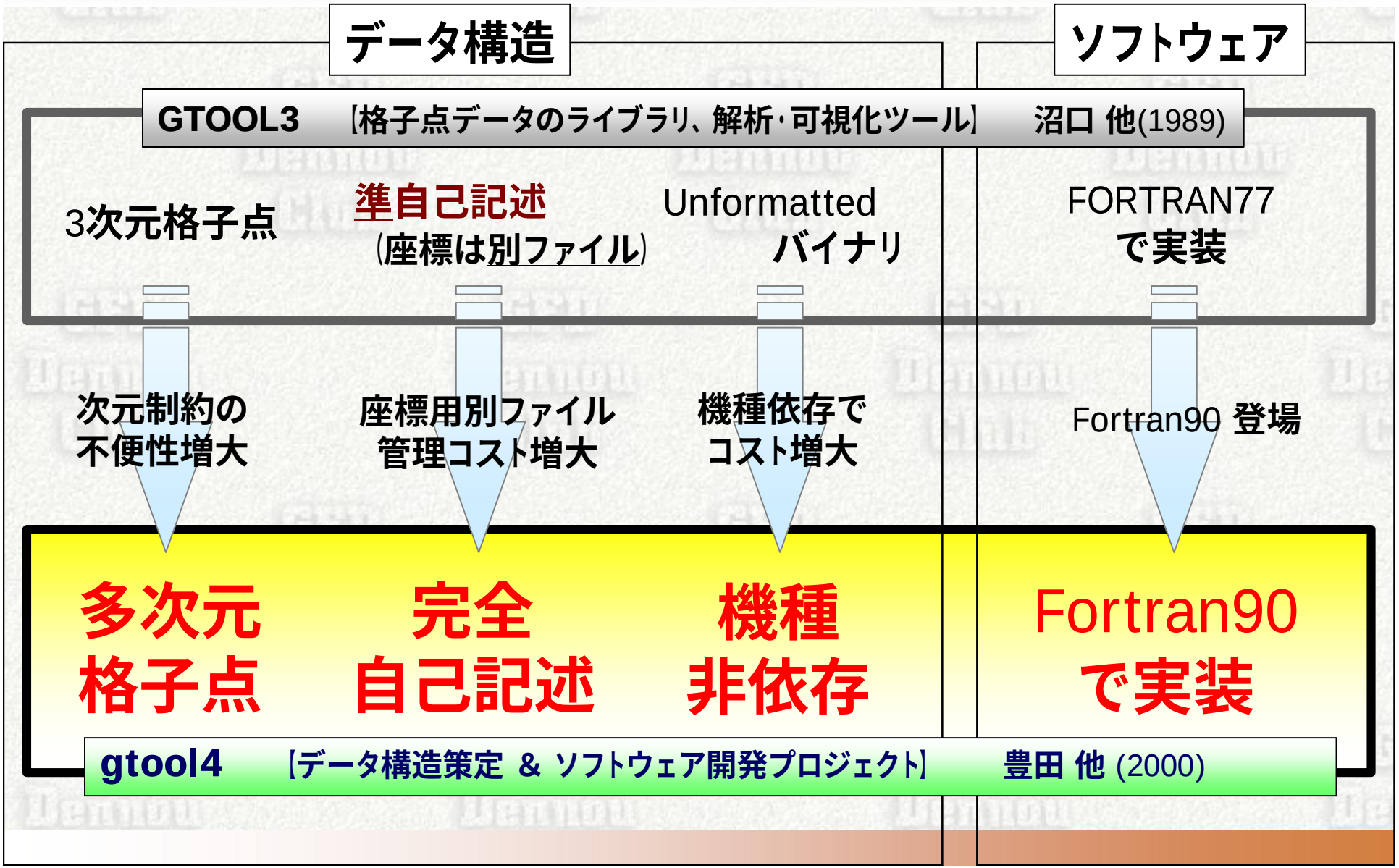
- 自己記述的データ構造
  - データに関する情報をデータに付属
    - 作成者・表題・履歴　　etc...
    - 変数・時刻・座標・欠損値・単位　etc...
- そのデータ構造のためのツール群
  - 数値モデル用データ I/O ライブラリ
  - 解析・可視化ツール

地球流体電脳倶楽部

gtool4 プロジェクト

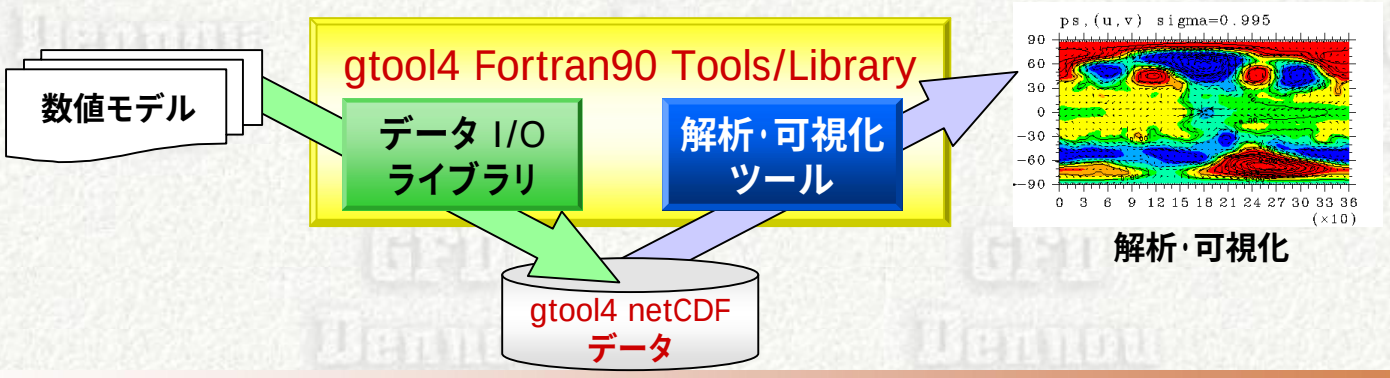
(1999〜2002 の軌跡)

gtool4 プロジェクト概要



「クリック一発で絵が書けるデータファイル」

- **データ構造　『gtool4 netCDF 規約』**
  - 多次元格子点データ
  - 完全自己記述的
    - 座標軸情報もデータに含む (軸データ不要)
    - 可視化情報もデータに含む (簡単描画が可能に)
  - 機種非依存
  - COARDS 規約、NCAR CSM 規約 との互換性を考慮
- **ソフトウェア『gtool4 Fortran90 Tools/Library』**
  - Fortran 90 ： モジュール、構造型、総称手続き を活用



gtool4 netCDF データ

gtool4 netCDF 規約に則った netCDF ファイルの例

```
$ ncdump sample.nc        (netCDF ファイルの属性 + データを出力)

[出力結果]
dimensions:
        x = 30 ;
        t = UNLIMITED ; // (201 currently)
variables:
        float x(x) ;
                x:long_name = "X-coordinate" ;
                x:units = "m" ;
                x:topology = "circular";
        float t(t) ;
                t:long_name = "time" ;
                t:units = "s" ;
        double temp(t, x) ;
                temp:long_name = "temperature" ;
                temp:units = "K" ;
                        :                              (続く)
```

次元変数

無制限次元
(主に時刻に使用)

変数

変数の
データ型

変数の属性
- long_name: 一般名称
- units: 単位
- :

gtool4 netCDF データ

gtool4 netCDF 規約に則った netCDF ファイルの例

```
(続き)                              :
// global attributes:
                  :title = "gt4_history sample" ;
                  :source = "Sample program of gt4_history/gt4f90io" ;
                  :institution = "GFD_Dennou Club davis project" ;
                  :history = "unknown > gt4_history: HistoryCreate\n",
      "" ;
data:
 x = 0, 0.03448276, 0.06896552, 0.1034483, 0.137931, 0.1724138,
0.2068965,
                    :
 t = 0, 0.0005, 0.001, 0.0015, 0.002, 0.0025, 0.003, 0.0035, 0.004,
0.0045,
                    :
 temp =
  1.38879542122922e-11, 3.87761921792298e-10, 8.53515772443671e-09,
                    :
```

大域属性 (ファイル全体の属性)

座標変数のデータ

従属変数のデータ

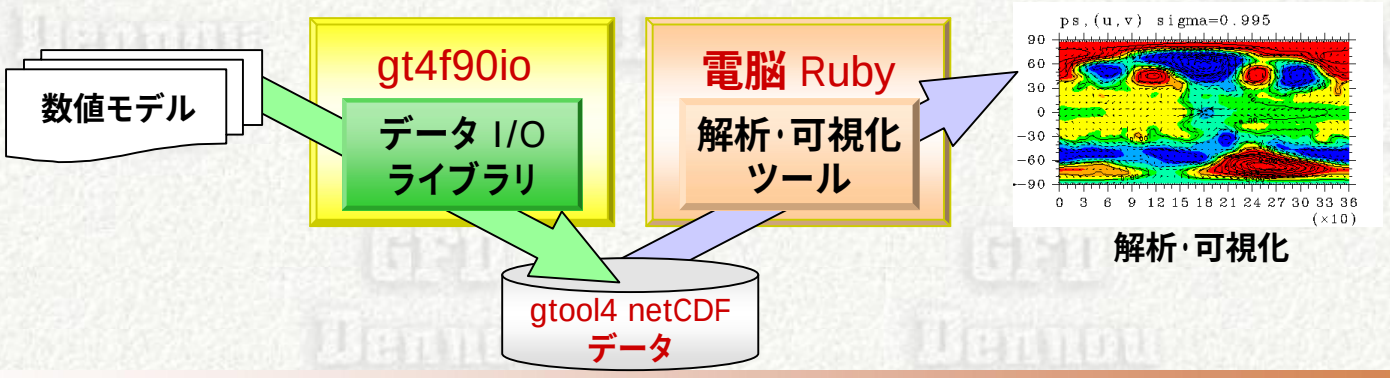
地球流体電脳倶楽部

gtool4 プロジェクト

(2003〜2006 の活動)

解析･可視化ツールは Ruby へ (2003)

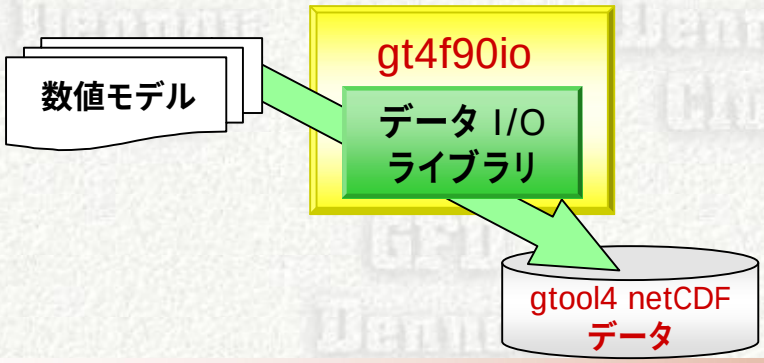
- **データ構造　『gtool4 netCDF 規約』**
  - **試用中 & 策定中**
- **ソフトウェア**
  - **解析･可視化ツール**: Dennou Ruby Project へ
    - **開発･保守･カスタマイズの手軽さ**　Ruby > Fortran
  - **データ** I/O: gt4f90io へ
    - **数値モデルとの親和性**
    - **I/Oとしての機能に特化**



gt4f90io

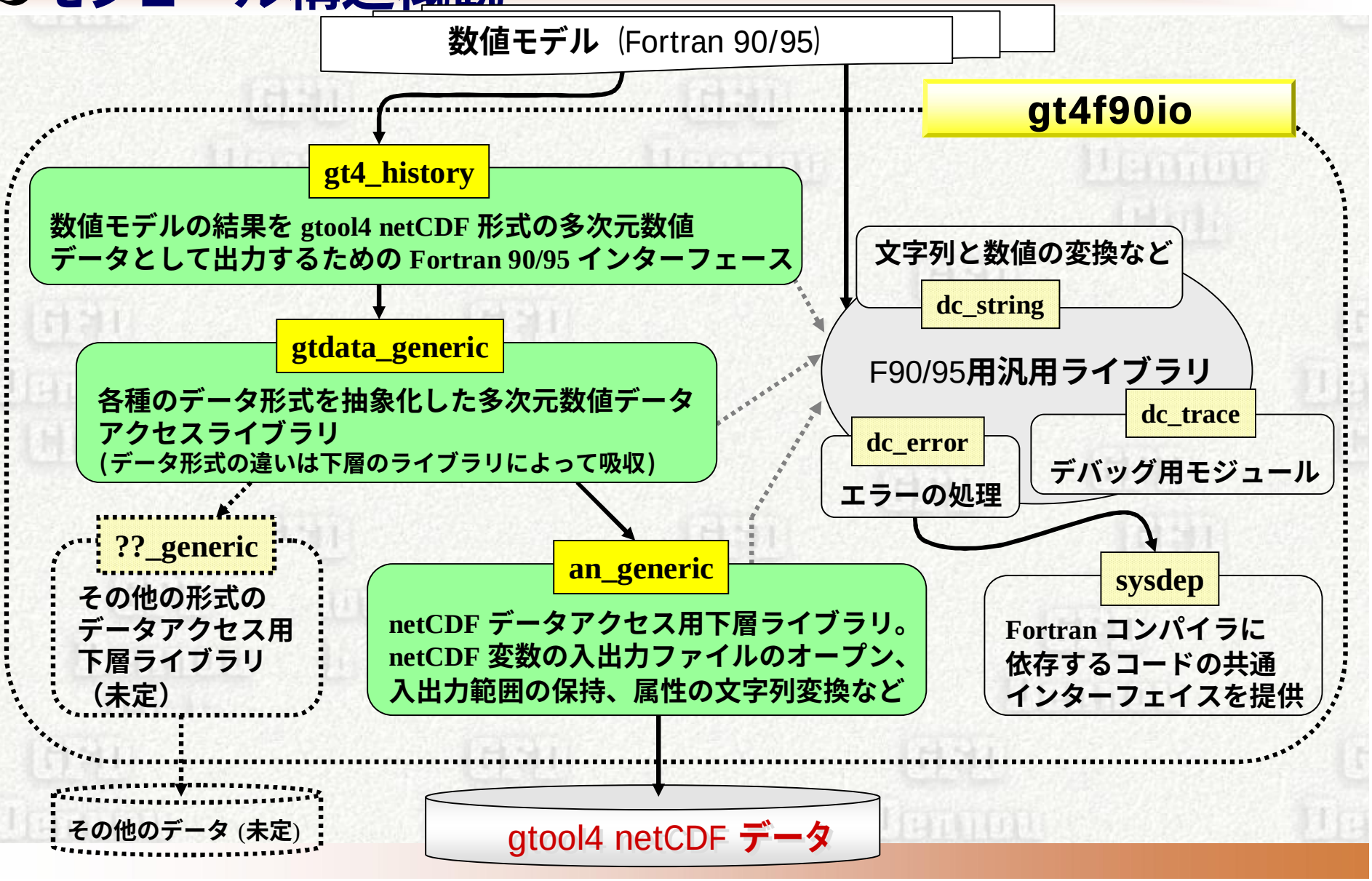
- **gtool4 netCDF データの I/O ライブラリ**
  - Fortran 90/95 **ベースの数値モデル用**

- **正式名称**
  - **[日]** gtool4 **規約に基づく** Fortran90 netCDF I/O **ライブラリ**
  - **[英]** Fortran90 netCDF I/O library with gtool4 convention



内部構造階層化

❍モジュール構造概観



お手軽インターフェース

○ライブラリ利用者は

- ❖ モジュール　gt4_history
- ❖ サブルーチン　5つ

を知っていれば OK

gt4_history のサブルーチン

- HistoryCreate(file, title, ...)
  - 初期設定
    - 出力ファイル名、タイトル、...、次元変数名、次元サイズ、...
- HistoryAddVariable(varname, dims, ...)
  - 変数定義
    - 変数名、依存次元名、...
- HistoryPut(varname, value, ...)
  - 変数出力
    - 変数名、出力値、...
- HistoryClose
  - 終了処理
- HistoryGet(file, varname, ...)
  - 変数入力
    - ファイル名、変数名、...

違うデータ型，次元数も
同サブルーチンで対応

使用例

❍サンプル Fortran 90 プログラム

```
program sample
  use gt4_history                                          ! モジュールの使用を宣言

  [型宣言] ......

  call HistoryCreate( &                                    ! ヒストリー作成
       file='sample.nc', title='gt4_history',    &         !    ・ファイル名、タイトル
       ..., dims=(/'x','t'/), dimsizes=(/30,0/), &         !    ・次元変数、次元サイズ
       .......)
  call HistoryAddVariable( &                               ! 変数定義
       varname='temp', dims=(/'x','t'/), .... )            !    ・変数名、依存次元、..

  [時間積分ループ]
         :
        call HistoryPut(varname= 'temp', value=temp) ! 変数の出力
         :
  [時間積分ループ 終わり]

  call HistoryClose                                        ! 終了の処理
  stop
end program sample
```

gt4f90io のご利益

- gtool4 netCDFデータを簡単に出力可能
- データ出力のための Fortran ソースコードの書式が揃う

プロジェクト進捗状況

gt4f90io 進捗状況 (2005〜)

- 新たなコンパイラ, アーキテクチャのサポート
  - G95, Fujitsu Fortran, Intel Fortran on Linux
  - NEC SX, HITACHI SR, FUJITSU VPP
- 入出力データの最大次元を3から7に拡張
  - パラメタスタディにも使いやすくなってきた...?
- リファレンスマニュアルは RDoc で
  - 常に最新のソースコードを反映
- F90 汎用ライブラリの充実
  - 文字列操作, デバッグ補助, 正規表現, コマンドライン引数解析, ハッシュの提供 (製作中)

gtoo4 規約 進捗状況(2006)

- 最近いろいろ検討中...
  - 可視化に関する属性の扱い
    - 全て Ruby ツールに任せ, gtool4 規約もソフトウェア同様身軽になる...?? (残しておいた方が良いのかも...)
  - CF 規約との比較
    - NetCDF Climate and Forecast (CF) Metadata Convention
      - 気象のシミュレーション業界で広まりつつあるらしい
      - CAM/CCSM, ECMWF ERA-40, FMS/GFDL などで利用されている. (他にも利用しているところがあったような...)
    - gtool4 ⇒ CF への移行は可能か?
      - CF は任意性が高すぎ, (自分たちが) 必要と思ういくつかの属性が無いため困難では...
    - CF との互換性は?
      - いくつかの属性に関しては互換性を図っても良いかも...

電脳 Ruby プロジェクト

- **解析, 可視化の道具立ては揃ってきてる**
  - **解析**: GPhys
  - **可視化**: RubyDCL, Ruby-VTK
  - GUI: gave
  - I/O: RubyNetCDF
- **データベース化の試み**
  - Ruby on Rails **を用いたデータベース作成**
    - **モデルデータ, 観測データ, 数値モデル, 解析・可視化スクリプトなどなどをデータベースに放り込む**
    - **検索や解析を** Web **ベースで**

まとめ

- 地球流体電脳倶楽部 gtool4 プロジェクト
  - 製品
    - gtool4 netCDF 規約
    - F90 データ I/O ライブラリ gt4f90io
  - 電脳 Ruby プロジェクトと一緒に解析・可視化ツール開発
- gt4f90io
  - 階層的モデル群の共通 I/O ライブラリとしてはだいたい出来上がってきた…?
- URL
  - http://www.gfd-dennou.org/library/gtool4

参考文献

- 地球流体電脳倶楽部 gtool4 プロジェクト
  - http://www.gfd-dennou.org/library/gtool4
- 電脳 Ruby プロジェクト
  - http://ruby.gfd-dennou.org/
- NetCDF Climate and Forecast (CF) Metadata Convention
  - http://www.cgd.ucar.edu/cms/eaton/cf-metadata/
- 規約の例 (Unidata 提供)
  - http://www.unidata.ucar.edu/software/netcdf/examples/files.html
- Community Climate System Model
  - http://www.ccsm.ucar.edu/models/ccsm3.0/
- The FMS Manual
  - http://www.gfdl.noaa.gov/~vb/FMSManual/
- dcmodel プログラミングガイドライン
  - http://www.gfd-dennou.org/library/dcmodel/coding-rules/dcmodel-coding-rules.htm
- DCPAM
  - http://www.gfd-dennou.org/library/dcpam/
- spml ライブラリ
  - http://www.gfd-dennou.org/library/spmodel/
- ISPACK
  - http://www.gfd-dennou.org/library/ispack/
- RDoc Fortran 90/95 解析機能強化版
  - http://www.gfd-dennou.org/library/dcmodel/

付録

おまけ（お手軽 Install）

❍debian パッケージあります

- ❖/etc/apt/sources.list に以下の 2 行を追加

```
deb ftp://www.gfd-dennou.org/library/cc-env/Linux/debian-dennou stable/
deb ftp://www.gfd-dennou.org/library/cc-env/Linux/debian-dennou stable/
```

- ❖apt でインストール

```
# apt-get install gt4f90io-g95
```

  - これはG95 Fortran 用。
  - 他にも Fujitsu Fortran, Intel Fortran 用のものも用意

❍使い方はチュートリアル参照してね

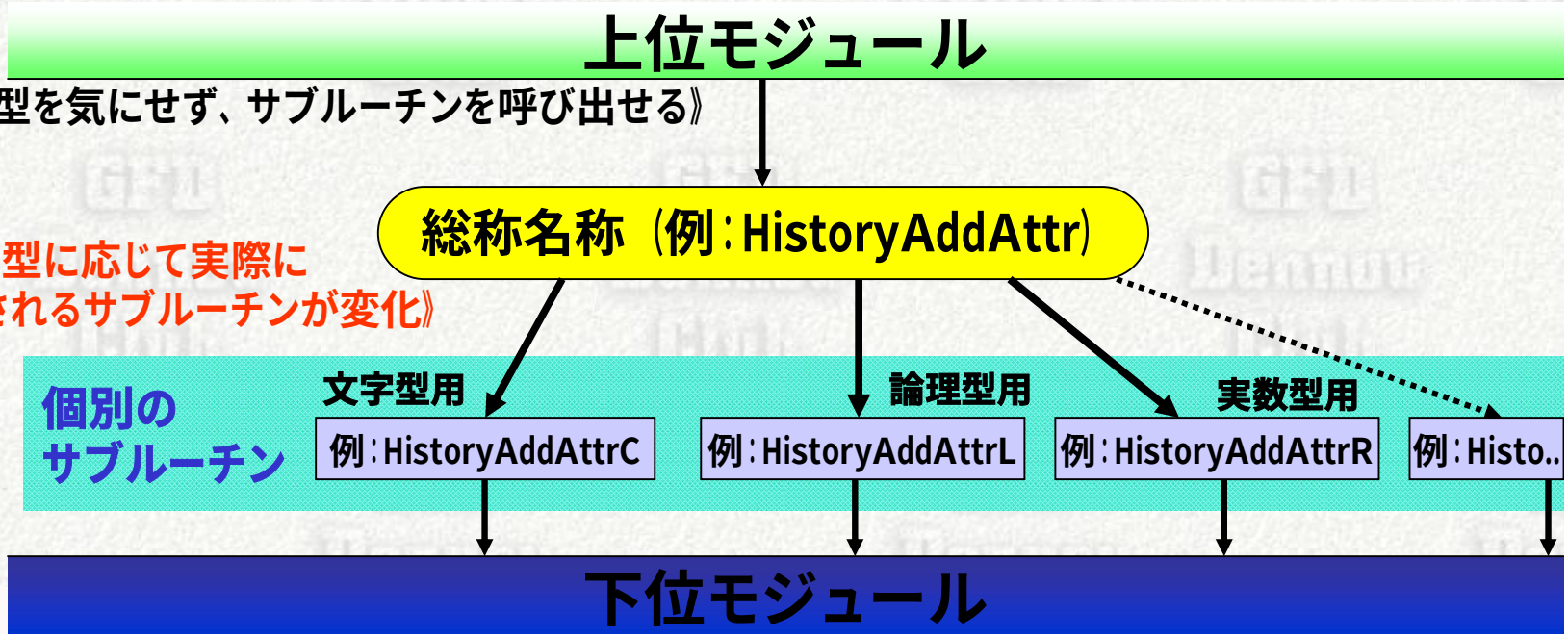
- ❖ http://www.gfd-dennou.org/library/gtool4/

総称手続き

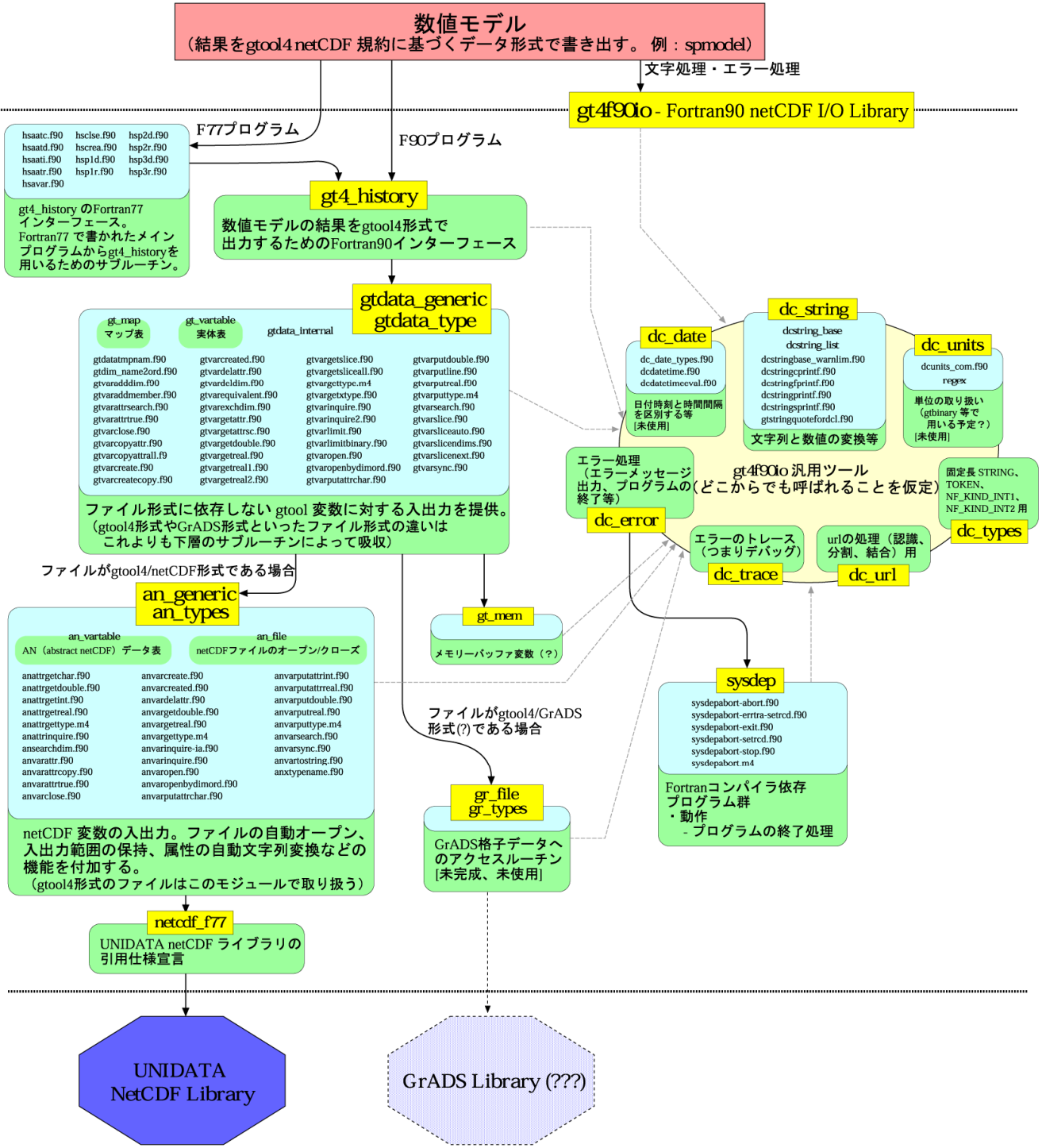
❍Fortran 90 コーディングスタイル

❖ オブジェクト指向“的”に…

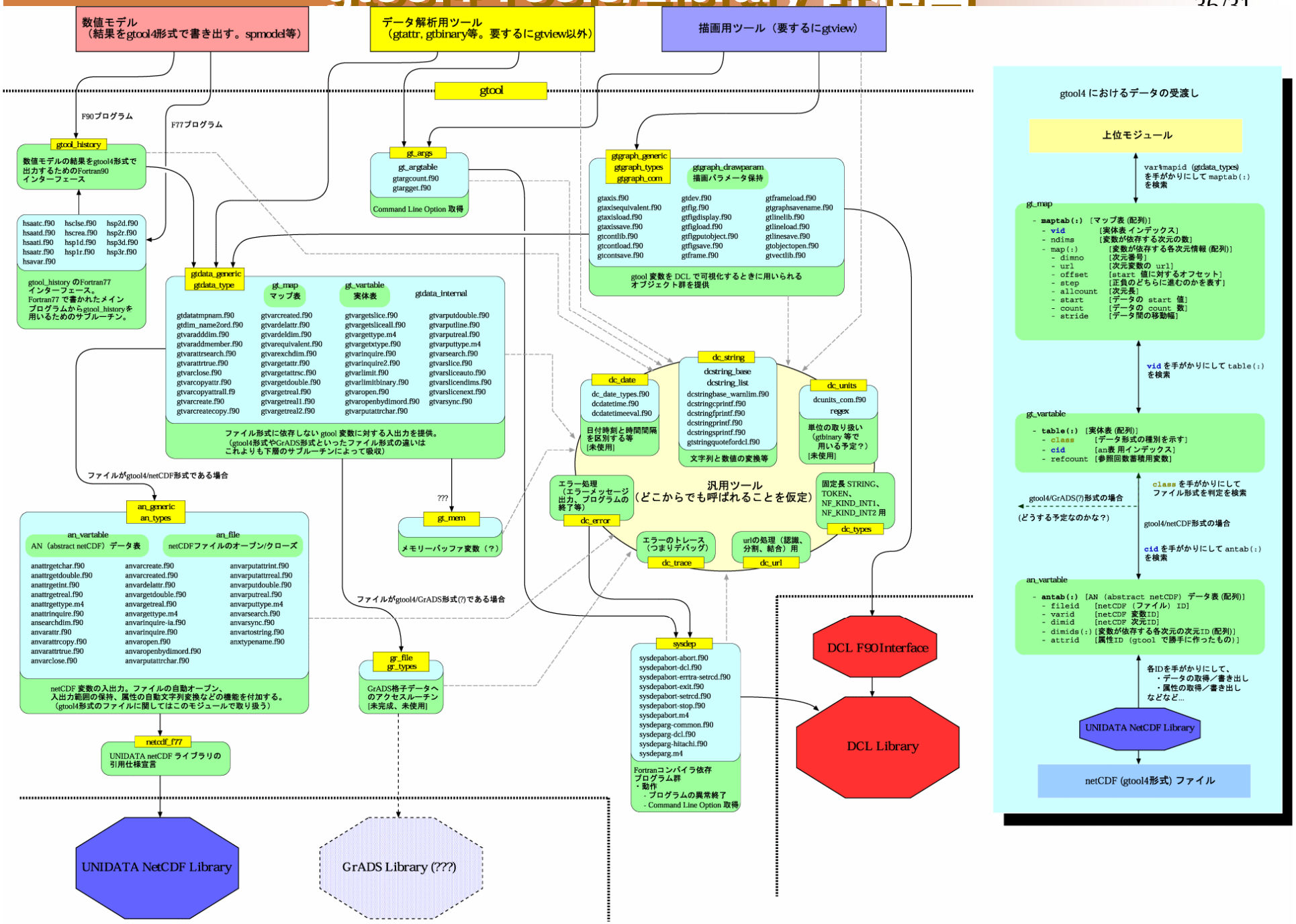
- クラス → 構造型
- メソッド → サブルーチン
- 多態性 (polymorphism) → 総称宣言されたサブルーチン
  - 引数の型に合わせ、異なるサブルーチンが呼び出される (下図参照)



gt4f90io 詳細図



gtool4 Tools/Library 詳細図



gt4_history 具体的使用例

❍サンプル Fortran 90 プログラム

```
program sample

  use gt4_history                                          ! モジュールの使用を宣言

  [型宣言] ..

  call HistoryCreate( &                                    ! ヒストリー作成
       file='sample.nc', title='gt4_history sample',    &  !    ・ファイル名の指定、データ全体の表題の指定
       source='Sample program of gt4_history/gt4f90io', &  !    ・データを生成する手段
       institution='GFD_Dennou Club davis project',     &  !    ・ファイルを最終的に変更した人/組織
       dims=(/'x','t'/), dimsizes=(/30,0/),             &  !    ・次元変数、次元のサイズの指定
       longnames=(/'X-coordinate','time        '/),     &  !    ・次元の名前
       units=(/'m','s'/),                               &  !    ・次元の単位の指定
       origin=real(0.0), interval=real(0.005) )            !    ・時間の原点、出力時間間隔の指定

  call HistoryPut('x',x)                                   ! 変数の出力
  call HistoryAddAttr('x', 'topology', 'circular')         ! 変数に属性を追加

  call HistoryAddVariable( &                               ! 変数定義（属性指定）
       varname='temp', dims=(/'x','t'/), &                 !    ・変数名、依存する次元の指定
       longname='temperature', units='K', xtype='double')  !    ・変数の(長い)名前、単位、変数の型の指定

  [時間積分ループ]
        :
       call HistoryPut('t',real(it*dt))                    ! 変数の出力
       call HistoryPut('temp',temp)                        ! 変数の出力
        :
  [時間積分ループ 終わり]

  call HistoryClose                                        ! 終了の処理
  stop
end program sample
```

gt4_history 使用結果

❍gtool4 netCDF 規約に則った netCDF ファイル

```
$ ncdump sample.nc          (netCDF ファイルの属性 + データを出力)
[出力結果]
dimensions:
        x = 30 ;
        t = UNLIMITED ; // (201 currently)
variables:
        float x(x) ;
                x:long_name = "X-coordinate" ;
                x:units = "m" ;
                x:topology = "circular";
        float t(t) ;
                t:long_name = "time" ;
                t:units = "s" ;
        double temp(t, x) ;
                temp:long_name = "temperature" ;
                temp:units = "K" ;

// global attributes:
                :title = "gt4_history sample" ;
                :source = "Sample program of gt4_history/gt4f90io" ;
                :institution = "GFD_Dennou Club davis project" ;
                :history = "unknown unknown> gt4_history: HistoryCreate\n",
    "" ;
data:
 x = 0, 0.03448276, 0.06896552, 0.1034483, 0.137931, 0.1724138, 0.2068965,
                :
 t = 0, 0.0005, 0.001, 0.0015, 0.002, 0.0025, 0.003, 0.0035, 0.004, 0.0045,
                :
 temp =
  1.38879542122922e-11, 3.87761921792298e-10, 8.53515772443671e-09,
                :
```

解析 and 可視化

by

RubyNetCDF
+ RubyDCL
+ Gphys
+ …