



# 気象庁におけるドキュメント

気象庁予報部数値予報課  
数値予報班 基盤整備グループ  
平原洋一

# 本日の内容

- ドキュメント

- ここでは開発・管理に必要な情報・文書とします。

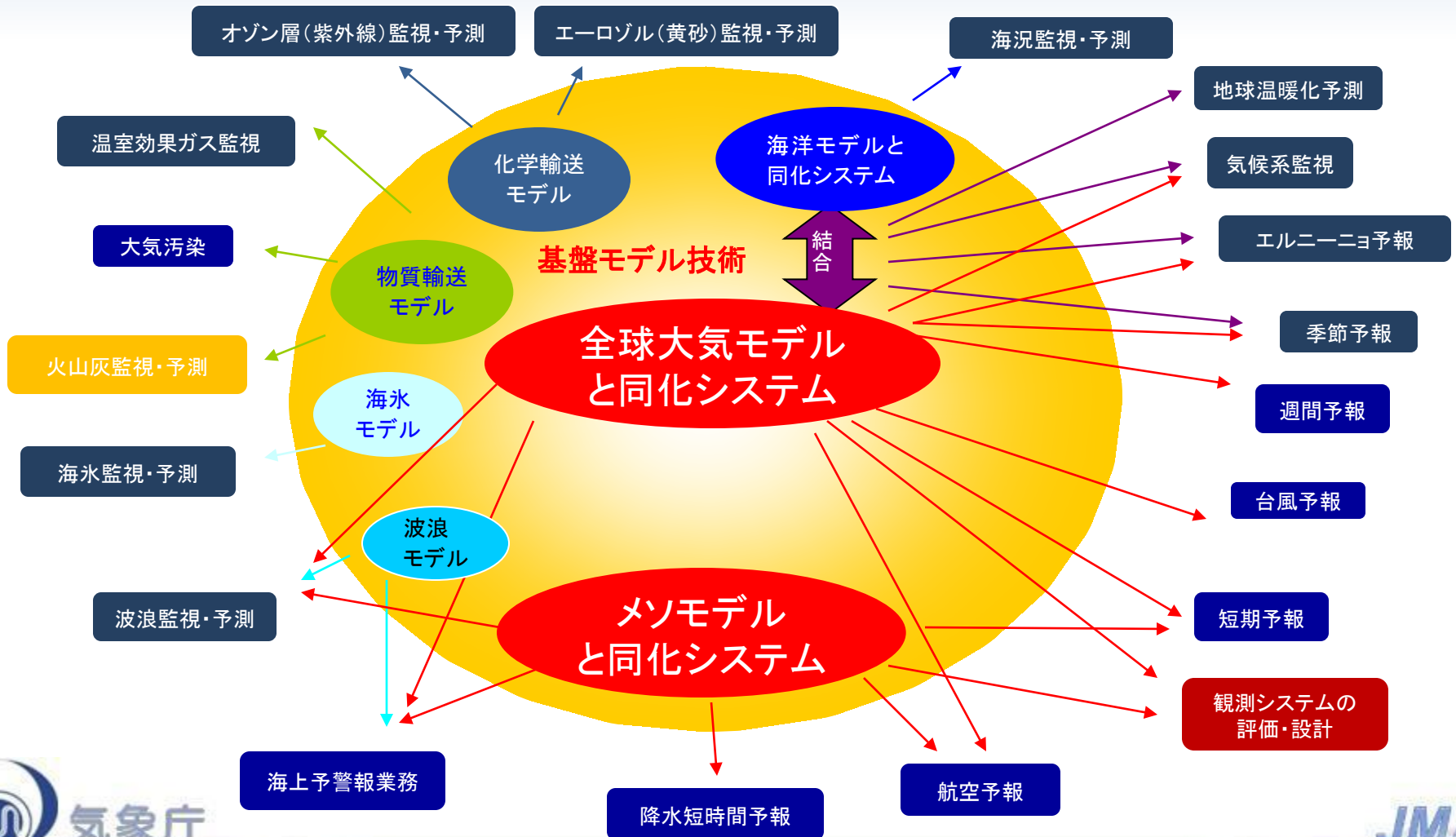
- ターゲット

- 「数値モデル」を含む、多くのプログラムが日々実行される「数値予報ルーチン」周辺に焦点をあてます。

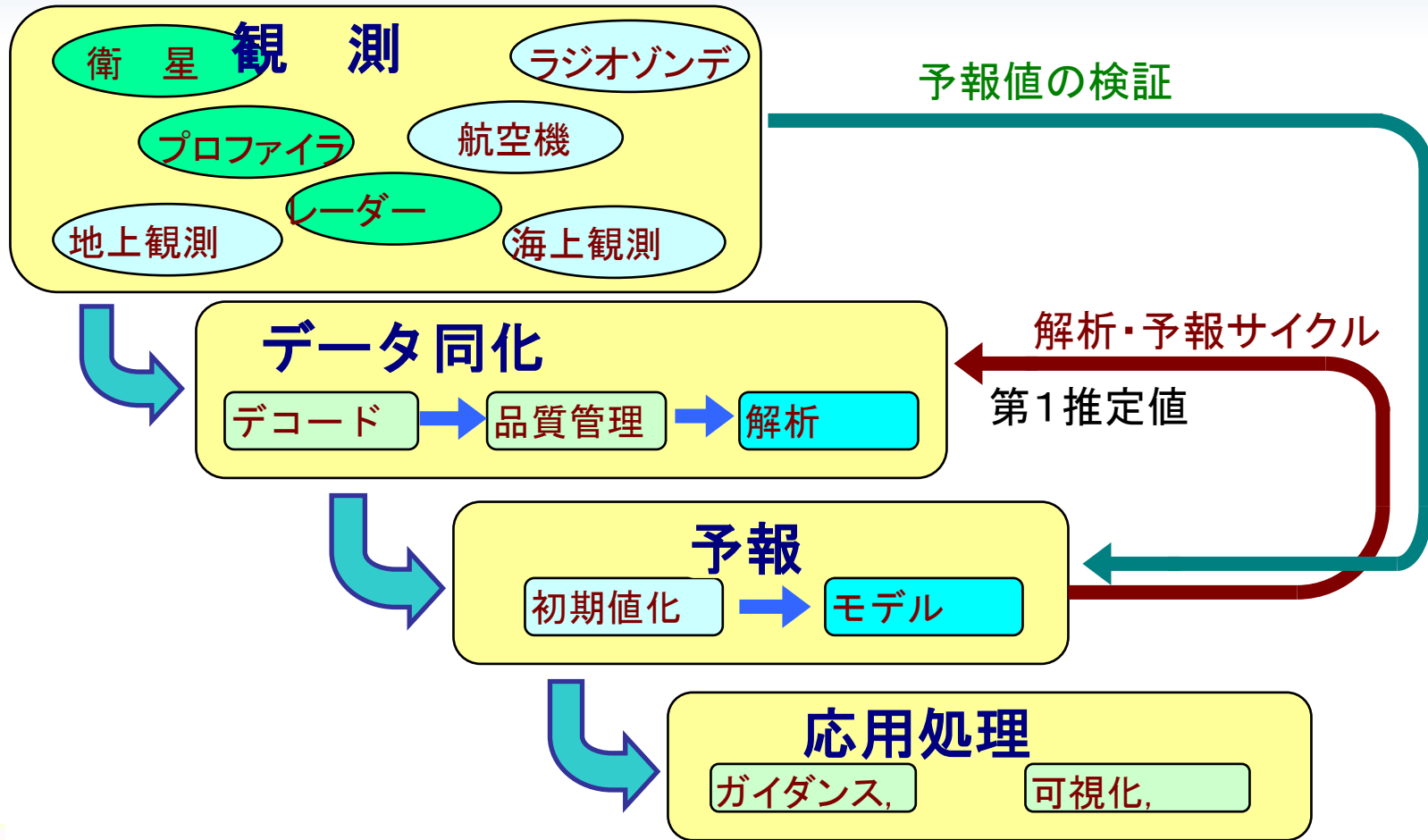
# 項目

- 数値予報ルーチン周りの概要
- 開発を行うにあたってのドキュメント
- ルーチン組み込み、運用にあたってのドキュメント
- 文書生成ツール活用の試み

# 数値予報モデルが支える 気象庁の業務



# 数値解析予報システム



# 数値予報ルーチン

- スーパーコンピュータシステムで実行
- 数値解析予報処理、関連処理による定型処理
- 高頻度から長期まで依存関係をもって日々実行
  - 毎時大気解析(1日24回実行)
  - 暖・寒候期予報(複数日に分けて実行)
- その規模
  - 実行プログラム 1,800個
  - バッチジョブ用シェルスクリプト 17,000個(880万行)
  - ソースファイル 20,000個(710万行)

# 開発からルーチン運用まで

## 開発

## 組み込み

## 運用

開発

評価検証

承認

ルーチン  
変更申請

審査・受付

組み込み  
試験

インストー  
ル

ルーチン運  
用の実施

各課室の開発担当

数値予報課プログラム班  
(ルーチン維持管理担当)

システム  
運用室

障害発生!!

→要対応

# 開発を行うにあたっての指針文書

- 「数値予報ルーチン申請の手引」
  - 数値予報課プログラム班 → 開発者
    - 数値予報ルーチンジョブの開発指針を定める
    - 申請手順
    - 禁止事項・留意事項（計算結果の再現性確保など）
    - 開発言語の指定（C、Fortran、Korn Shell、Ruby、AWK）
    - 新規に作成するFortranプログラムについては、標準コーディングルールに準拠することを推奨
      - <http://www.mri-jma.go.jp/Project/mrinpd/coderule.html>
      - 開発プロジェクトによっては標準コーディングルールを基本とし、さらにローカルなスタイルルールを定める場合も。



# 開発

- 共同作業であり、担当者は変わりうるもの。
- 管理すべき対象も多岐にわたる。
  - ソース、ビルド情報、定数、ジョブ実行用ファイル・・・
- 情報共有、引き継ぎ等に文書化は不可欠。
  - 保守・開発者向け文書
    - ソースコード解読・改修に必要な情報
  - ユーザー向け文書
    - モデル・ツールの利用方法、ライブラリリファレンス

# 保守・開発者向け文書

- プログラムの寿命は長い
  - 実際の開発では既存のプログラム改修が多い。
  - 古いソースは相当古い。
    - FORTRAN77も普通に存在。
- 担当が変わっても支障が生じないように、必要な情報を文書化(改修の手掛かり)
  - 処理概要、処理の流れ、影響範囲、変更理由・履歴等
  - 従来よりソースコメント、別ファイル文書、書籍、メール、イントラネット、Wiki 等々...

# 保守・開発者向け文書

- 近年はバージョン管理システムの活用事例が増加
  - Subversion、Git
  - 変更履歴の手作業による管理から脱却
- 一部の大規模プログラム(モデル等)に関してはプロジェクト管理システムの導入が進む
  - Trac、Redmine
  - よりシステム化された変更履歴の管理

# 保守・開発者向け文書

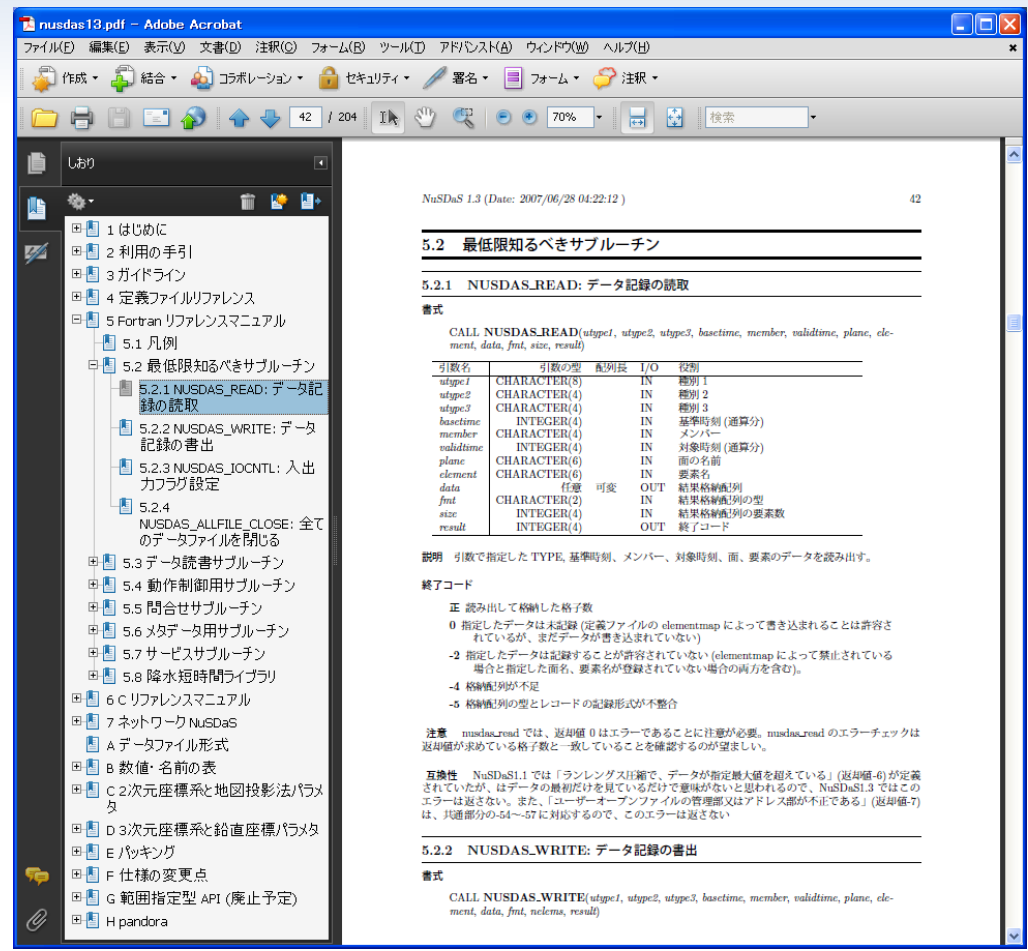
- 文書生成ツールの活用は？
  - Rdoc, Doxygen 等々
  - 本格的に利用されている例はまだまだ少ない
    - ソースが古い場合は辛い(FORTRAN77・・・)
    - コメントに日本語を使用できない場面も(移植性への影響等)
  - ライブラリリファレンス作成には比較的相性がよいため活用事例も

# ライブラリリファレンス

- 開発者 → 他の開発者、ユーザー
- ライブラリ例
  - 数値予報標準ライブラリ
    - 物理要素変換・地図投影・システムユーティリティ
  - NuSDaSライブラリ
    - NuSDaS: NWP Standard Dataset System
  - その他
    - 特定フォーマットのデータ操作(デコードデータ等)
    - Fax描画用 等々

# 例: NuSDaSDドキュメント

- ソースコードからの文書生成ツールを活用
- 文書生成ツールは独自整備
  - TeXファイルに、抽出したソースコメントを埋め込む
- PDF作成までの定型処理を自動化



(気象研究コンソーシアムHPより)

<http://www.mri-jma.go.jp/Project/cons/nusdas13.pdf>

# 組み込みにあたっての文書

- ルーチン変更申請書

- 開発者 → 数値予報課プログラム班

- 変更内容の基本情報を記述したシート(Excel)
  - 変更概要、動作試験での要確認事項、実体のリスト
- 実体もあわせて提出

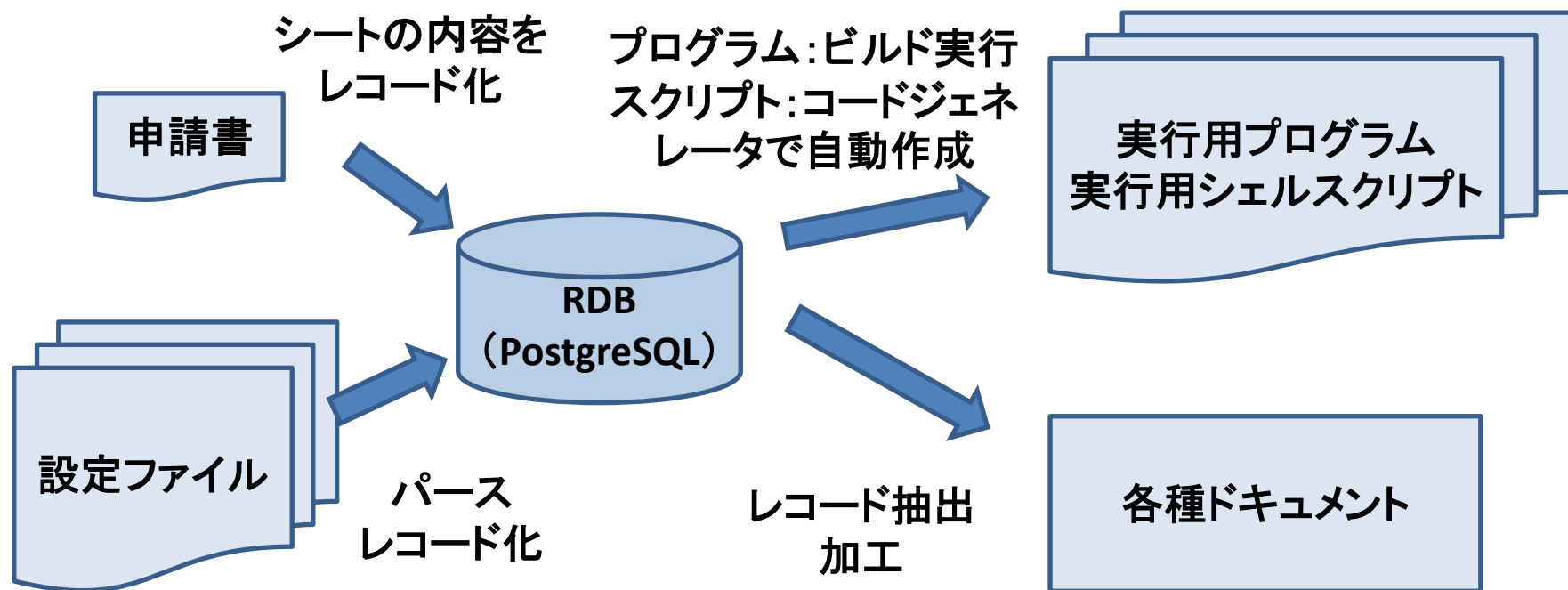
- ソースファイル
- 定数データ

- ジョブ制御ファイル(世間一般のJCLのようなもの)
- ビルド情報記述ファイル(メイクファイルのようなもの)
- データ定義ファイル(概要等を記述した簡単なリスト)
- ジョブ定義ファイル(ジョブ実行順序、資源を記述)

独自整備の  
設定ファイル

# 組み込み

- ルーチン情報はRDBで一元管理
  - 機械的なチェック、インストール、ドキュメント作成





# RDBを活用したドキュメント例

保存データ一覧

ジョブネット図

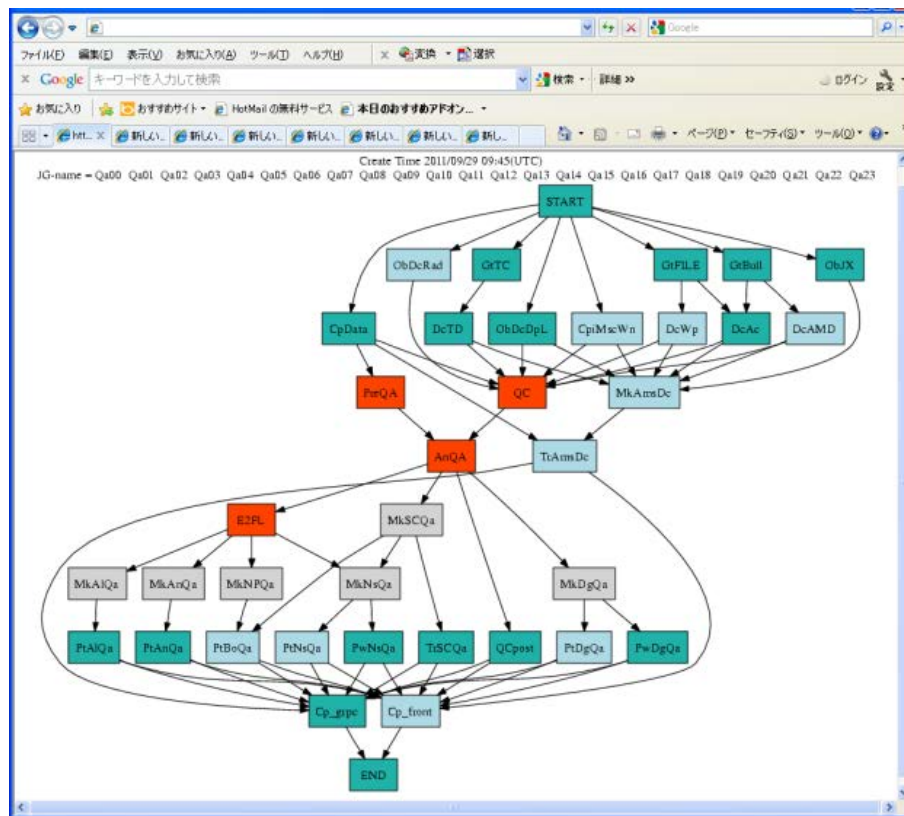
ctf-data - Windows Internet Explorer

ctf-data

**/Mf/Fcst/fcst\_p.nus**

可変データ分類名	/Mf/Fcst/fcst_p.nus
データ概要	メノ気圧面予報値
申請データサイズ	4280000KB
長期保存期間	3年保存
CTLパス	/ctl/nwp_r2/3years/Mf/Fcst/fcst_p.nus

データ作成JCL(ルーチン運用中のもの)	データ作成Sh(ルーチン運用中のもの)
	/Mf/MF00/MFc.UseEF
	/Mf/MF03/MFc.UseEF
	/Mf/MF06/MFc.UseEF
	/Mf/MF09/MFc.UseEF
	/Mf/MF12/MFc.UseEF
	/Mf/MF15/MFc.UseEF
	/Mf/MF18/MFc.UseEF
	/Mf/MF21/MFc.UseEF
/Mf/Comm/MFc.UseEF.jcl	
	/Mf/MF00/MFc
	/Mf/MF03/MFc
	/Mf/MF06/MFc
	/Mf/MF09/MFc
	/Mf/MF12/MFc
	/Mf/MF15/MFc
	/Mf/MF18/MFc
	/Mf/MF21/MFc
/Mf/Comm/MFc.jcl	



↑ Graphviz活用: 依存関係の可視化

# 運用にあたっての文書

- 数値予報ルーチン運用マニュアル
  - 数値予報課プログラム班 → システム運用室現業
    - 数値予報ルーチン概要・実行環境
    - 障害対応手順
    - 付属資料
      - ジョブ一覧
      - 障害対応一覧
      - プロダクト一覧
      - その他(タイムスケジュール表など)

一部RDBを活用

# 今後に向けての取り組み

- 高度化・多様化する業務
- 複雑化するシステム
  - 情報共有(文書化)の推進
  - 効率的な枠組み
    - 自動化できるところは自動化
    - 開発 → 運用の連携強化(作業コスト削減)



# 例：文書生成ツールの活用

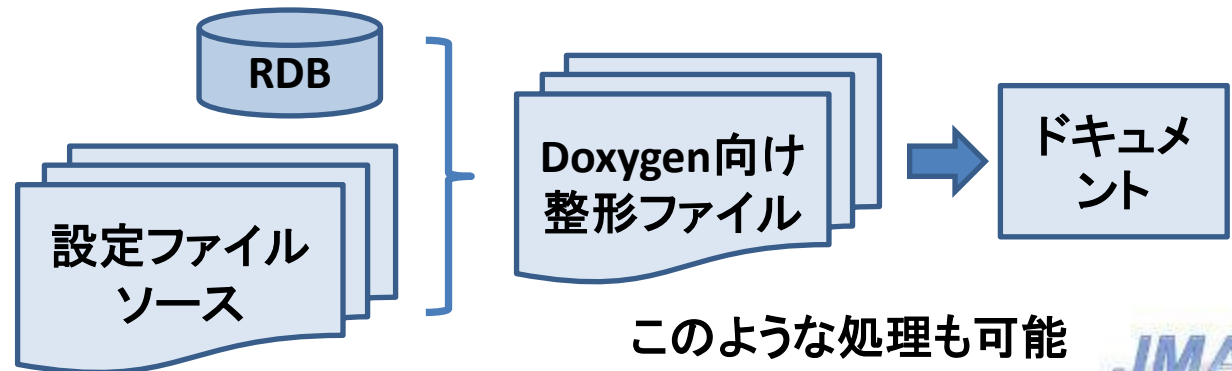
- 例えばDoxygen

- 近年、Fortranにも対応

- FORTRAN77はいろいろ難あり...

- 柔軟性に富む

- 文書化されたソースファイルだけでなく、工夫次第で  
**様々なフォーマットの情報をもとに、オンラインドキュメントを生成可能**



このような処理も可能



# 例：文書生成ツール活用の試み

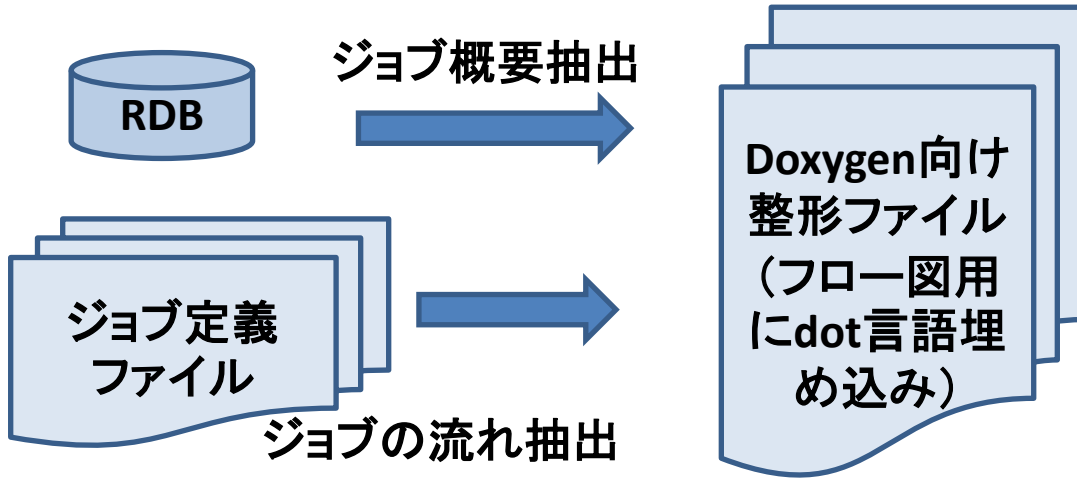
- ソースに限らず、開発者が関知すべき情報を構造的にドキュメント化
  - 全体的な処理の流れ(ジョブフロー)
  - ジョブの構成(プログラム実行順序)
  - プログラム実行情報(入出力データ情報等)
  - 実行プログラム情報(ソースファイル等)
  - ソースファイル情報(関数・サブルーチン)

– ツールを利用して効率的に



# 例：文書生成ツール活用の試み

## ・ジョブフロー



## 例：全球地上解析(Aa)

**Aa**

メインページ    ファイル   

**Aa ドキュメント**

**JG-names:**

- Aa00 Aa03 Aa06 Aa09 Aa12 Aa15 Aa18 Aa21
  - 天気図解析(ASAS)の基礎資料となる地上解析値の作成JG

**日付:** Create Time 2011/08/14 22:42(UTC)

**JOBS:**

- START
- CpData : 第一推定値の別JG(EF)からの取得
- GtDcBL : 必要デコードデータ取得
- QcAnal : 観測データの品質管理の実行、2次元最適内挿法による地表面解析実施、および解析値の等緯度経度格子への変換。
- CpoFcdAa : 予報課業務用サーバーに解析結果のコピー
- Cp\_front
- Cp\_grpc
- END

START  
↓  
CpDat a  
↓  
Gt DcBL  
↓  
QcAnal

次ページ →

```
@JOB,GtDcBL,Comm,RCX,0,00:05:00,continue,,/,grpK/nwp/Rtn/Comment/Aa/GtDcBL
@Prec_Job,CpData
@End_Status,0-0:NORMAL,1-255:ABEND,
@Script_select,NORMAL::,TOKU01:DZERO:,
#####2
```

```
@JOB,QcAnal,Comm,RHX,1,00:05:00,continue,,/,grpK/nwp/Rtn/Comment/Aa/QcAnal
@Prec_Job,GtDcBL
@End_Status,0-0:NORMAL,1-255:ABEND,
@Script_select,NORMAL::,TOKU01:DZERO:,
@JOB_Env,jnos_resources:ConsumableCpus(16)
```

(...以下省略)

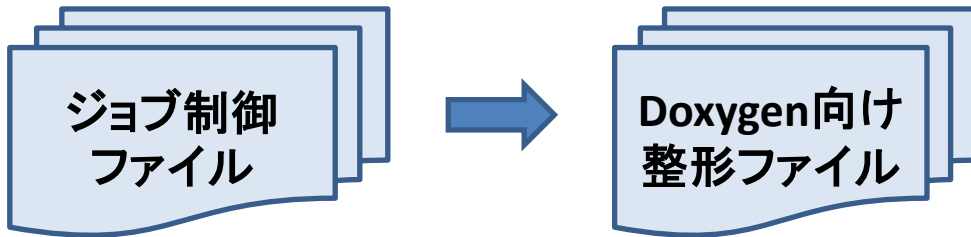
## ジョブ定義ファイル例



# 例：文書生成ツール活用の試み

## ・ジョブの構成

プログラム実行ステップの抽出



job Aa:QcAnal

### ジョブ制御ファイル例

step qc

```
pgm jpp:/exe/Anal/Qcxa/aaqc.sxx
  dd in name=PRM literal=<<END_OF_PRM
&qcflag int=T, trk=T, ext=T, gross=T, space=T /
END_OF_PRM
  dd in name=HEADER      data=jpp:/const/Anal/Qc/header_ga.txt
  dd in name=DCDTHIN_FT.01
data=jpp:/const/Anal/Qc/Namelist/dcdthin_aa.txt
  dd in name=DCDTHIN_FT.02 data=jpp:/jgdir/timecard.txt
  dd in name=DCDTHIN_FT.10 data=jpp:/jgdir/Dcd/Aa/surf_dcdf.dat
```

(・・・以下省略)

## 例：QC・解析ジョブ (QcAnal)

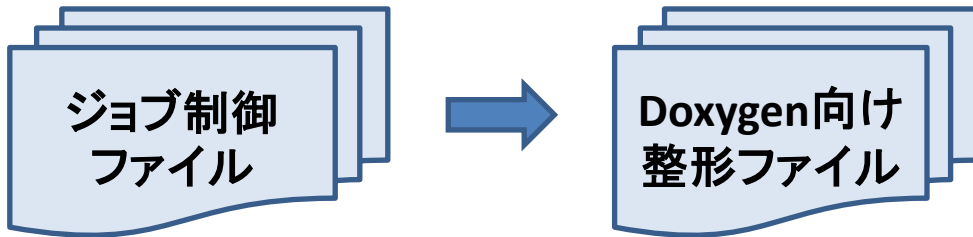
次ページ



# 例：文書生成ツール活用の試み

## ・プログラム実行情報

プログラム実行ステップの情報抽出



job Aa:QcAnal

### ジョブ制御ファイル例

```

step qc
pgm jpp:/exe/Anal/Qcxa/aaqc.sxx
  dd in name=PRM literal=<<END_OF_PRM
&qcflag int=T, trk=T, ext=T, gross=T, space=T /
END_OF_PRM
  dd in name=HEADER      data=jpp:/const/Anal/Qc/header_ga.txt
  dd in name=DCDTHIN_FT.01
data=jpp:/const/Anal/Qc/Namelist/dcdthin_aa.txt
  dd in name=DCDTHIN_FT.02 data=jpp:/jgdir/timecard.txt
  dd in name=DCDTHIN_FT.10 data=jpp:/jgdir/Dcd/Aa/surf_dcdf.dat

```

(・・・以下省略)

## 例：プログラム実行ステップ：qc

The screenshot shows the 'QcAnal' web interface. The 'Related Pages' tab is active, displaying details for step 'qc'. The 'Programs' section lists 'PGM : mod/Anal/Exe/Qcxa/aaqc.sxx', which is circled in red with an arrow pointing to the right. The 'Input Variables' and 'Output Variables' sections list various file paths and parameters. The 'Input Constants' section lists a large number of constants with their corresponding file paths. A '次ページ' (Next Page) button is visible in the top right corner.

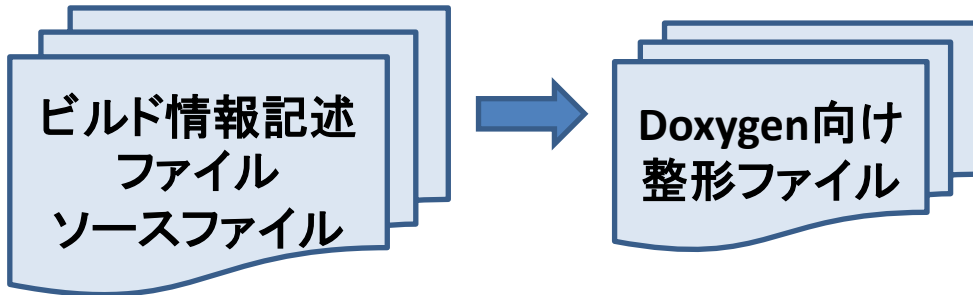




# 例：文書生成ツール活用の試み

## ・実行プログラム情報

コメント・ソースファイル情報の抽出



```

#% @mainpage
#% @par メインルーチン
#% unitedQC()
#%
#% @par 目的:
#% 観測データの統合品質管理
#%
#% @par 概要:
#% 観測データの統合品質管理(内的・外的QC)を実施し、解析処理に入力する
#%
#% @par 入力データ(変数):
#% - DCDTHIN_FT.02: タイムカード

```

ビルド情報記述ファイル  
に付加したDoxygen用  
コメント

(・・・以下省略)

例：実行プログラム：aaqc.sxx

The screenshot shows a web browser displaying the document 'aaqc.sxx'. The browser's address bar shows 'aaqc.sxx' and the page title is 'aaqc.sxx ドキュメント'. The content area shows the following information:

- メインルーチン:** unitedQC() (highlighted with a red circle and an arrow pointing to the right)
- 目的:** 観測データの統合品質管理
- 概要:** 観測データの統合品質管理(内的・外的QC)を実施し、解析処理に入力するCDAデータを生成する。
- 入力データ(変数):**
  - DCDTHIN\_FT.02: タイムカード
  - DCDTHIN\_FT.10: 地上観測デコードデータ
  - TIMECARD: タイムカード
  - CDATRK\_FT.61: QC蓄積ファイル
  - NUSDAS50: 第一推定値
- 出力データ:**
  - CDA1\_FT.61: QC蓄積ファイル(aqcmisc0.dat)
  - CDA1\_FT.62: QC蓄積ファイル(更新値)(aqcmisc.dat)

次ページ

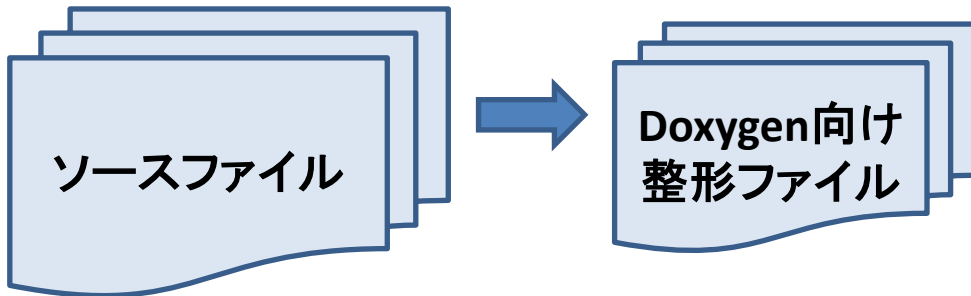




# 例：文書生成ツール活用の試み

## ・ソースファイル情報

ソースファイル情報の抽出



例：ソースファイル：qcmain.F90

aaqc.sxx

メインページ	モジュール	データ型	ファイル
🔍 検索			
ファイル一覧	ファイルメンバ	マクロ定義   関数/サブルーチン	
<b>qcmain.F90</b>			

ソースコードを見る。

マクロ定義

```
#define OLDCDA2
#define OLDCDA2
関数/サブルーチン
```

program	unitedQC
subroutine	unitedQC_FileName (filename, name, unitNo, rankNo)
subroutine	unitedQC_Open_Cda (unitNo, cda, mode, size, rankNo)
subroutine	unitedQC_Close_Cda (unitNo)
subroutine	unitedQC_Open (unitNo, step, rankNo, access, form, status, filestat)
subroutine	unitedQC_Open_txt (unitNo, step, rankNo, status, filestat)
subroutine	unitedQC_Open_dat (unitNo, step, rankNo, status, filestat)

マクロ定義

```
#define OLDCDA2
#define OLDCDA2
```

## ソースファイル例

```
program unitedQC
#ifdef RA20
#define OLDCDA2
#endif
#ifdef RA10
#define OLDCDA2
#endif
use cdaio, only : cdamovefilecontrolindex, cdarewind, &
                & convert_dcdasrt_to_cdasort, cdasort, &
                & typeSortInf, typeSelectInf
implicit none
#ifdef MPI
include 'mpif.h'
#endif
integer, parameter :: dcdf = 10000000, &
                    & cda0f = 50000000, &
```

(・・・以下省略)

# 参考

- Fortran標準コーディングルール
  - <http://www.mri-jma.go.jp/Project/mrinpd/coderule.html>
- NuSDaS
  - <http://www.mri-jma.go.jp/Project/cons/nusdas13.pdf>
- Doxygen
  - <http://www.stack.nl/~dimitri/doxygen/index.html>
- Graphviz
  - <http://www.graphviz.org/>