

ソースコードとリファレンスマニュアルの 作成方法について

～ RDoc Fortran 90/95 ソースコード解析機能強化版 の紹介 ～

佐々木洋平

京都大学・数学教室

uwabami@gfd-dennou.org

地球流体データ解析・数値計算ワークショップ
～ 数値モデル開発のための基盤技術と新しい解析可視化手法 ～

2011 年 10 月 7 日, 惑星科学研究センター

- 1 話の「まくら」
- 2 リファレンスマニュアルの作成アレコレ
- 3 RDoc-f95
- 4 今後の展望/議論の種

1 話の「まくら」

2 リファレンスマニュアルの作成アレコレ

3 RDoc-f95

4 今後の展望/議論の種

自己紹介/disclaimer

自己紹介

名前 佐々木 洋平

- 2010 年に学位取得→京大へ
- 専門: 回転球殻対流/MHD ダイナモの数値計算

趣味 **FLOSS 開発**

- Debian, TeXLive, GCC, Emacs, Ruby...

disclaimer

- 今日は FLOSS の開発者, の気分でお話しします
 - 「地球流体電脳倶楽部」での「文書作成」という話は半分くらい
- 本発表の内容は個人的な見解です
 - 「地球流体電脳倶楽部/dcmodel プロジェクト」の見解ではありません

ソースコードと「ドキュメント」

- ドキュメントの重要性
 - 第三者への提供 (プログラムの利用)
 - 開発や保守の効率化 (プログラムの改変)
- 「ドキュメント?」
 - 数理・離散化に関する文書
 - コードのリファレンスマニュアル

リファレンスマニュアル?

- コードを「読む」際に必要となる情報が書かれている文書
- 具体的には?
 - ディレクトリ構造, ファイル構成, 命名規則, 略語, 歴史...
 - データ構造
 - 『... プログラムに解説をつけるための、もっとも効果的な方法の一つは、単にデータの割り付けかたをくわしく説明する、というものである。おもな変数について、その値としてはどんなものが可能かを示し、それが変わっていくようすを説明すれば、それだけでプログラムの解説は、ずいぶん進んだといつてよい。...』(プログラミング書法, pp.168)
 - 関数同士の呼び出し関係 (call graph)
 - 関数そのもの。
 - 数値モデルの場合は main がある (筈な) のでそこから読む, など
- どうやって, そのようなリファレンスを作成するか?

① 話の「まくら」

② リファレンスマニュアルの作成アレコレ

③ RDoc-f95

④ 今後の展望/議論の種

リファレンスマニュアルの作成ツール

- 大きく二通り
- ソースコード埋没型: コードにドキュメントを埋め込む
 - インターフェース部分に記述 or 関数部分に記述
 - 実装: Doxygen, Javadoc, GTK-Doc, RDoc, YARD, ...
- ソースコード非埋没型: ドキュメントは別途作成
 - ドキュメント量が多い場合, 公開したい API を厳密に選別したい場合, 等
 - 実装: Sphinx, BitClust, Texinfo, RDTTool, ...

ソースコード埋没型

- メリット
 - ソースコードの近くにドキュメントがある→変更漏れの防止
 - 確実に防げるわけではないが
 - ドキュメントを書いていない場合もそれっぽくドキュメントができる
 - プログラムをパース→変数, 関数を自動収集するから
- デメリット
 - ソースコード内にコードよりドキュメントの方が多くなったりする
 - 可読性が悪い(と感じるかも)
 - (自動収集により) 意図しない API がドキュメントされてしまうことも
 - ツールだけでは, 複数の (自然) 言語に対応できない
 - 少なくとも現実的ではない

ソースコード非埋没型

- メリット
 - 公開したい API だけを選別して公開できる
 - 複数の（自然）言語をサポートしやすい
- デメリット
 - ソースコードの更新に追従するために変更分を確認するツールが必要
 - ドキュメント化する API を収集するツールが必要

どうすべきか？

- 特にこだわりが無いなら, 埋没型が良い
 - コード内にドキュメントをだらだらと書く必要は (あんまり) 無い
 - 意図しない API の公開はコードが悪い (ことが多い)
 - 多言語対応は面倒 (後述)
- 非埋没型が得意な分野は結構限られる
 - 複数のドキュメントをまとめる場合, 言語そのものを記述する場合
 - 大規模モデルのトップは, 埋没型と組み合わせるのが良いかも
 - 多言語対応を考える場合

埋没型での多言語対応?

- 言語タグに対応するツールが無い... ような
 - Doxygen ではコレできらしい
 - 詳細は河合さんの発表にて
- en, ja しか無い世界なら, それでも良いかもしれないが.
 - 言語が 3 つ以上になると破綻.
 - 「コードとドキュメントの乖離を防ぐ」というメリットが消える

- 1 話の「まくら」
- 2 リファレンスマニュアルの作成アレコレ
- 3 RDoc-f95
- 4 今後の展望/議論の種

RDoc Fortran 90/95 ソースコード解析機能強化版

RDoc: Ruby のリファレンス生成ツール

URL <http://rdoc.rubyforge.org/>

Copyright Dave Thomas, Eric Hodel, Ryan Davis

License GPL or Ruby's

特徴 埋没型. シンプルなタグ付け. Ruby, C に対応. parser を書けば他の言語にも対応可能

RDoc-f95: fortran 90/95 の parser + 本体の拡張

URL <http://www.gfd-dennou.org/arch/dcmmodel/rdoc-f95/>

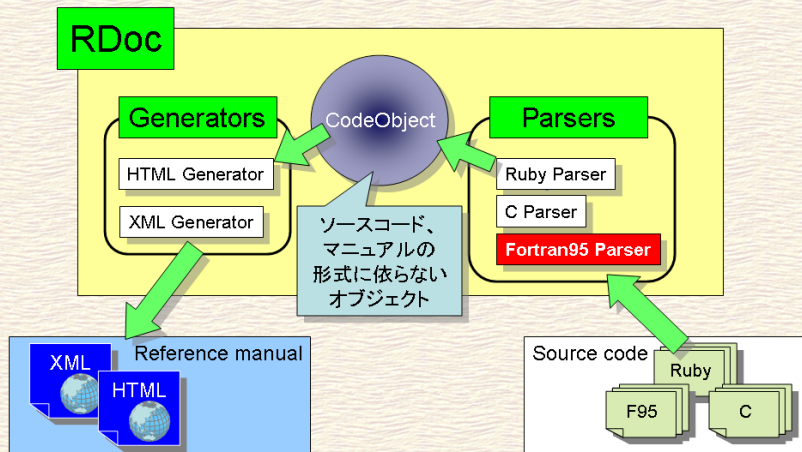
Copyright Yasuhiro Morikawa(? 明記してないですね).

License 本家 RDoc と同じ → GPL or Ruby's

特徴 fortran90/95 parser のコード自動解析強化

RDoc によるドキュメント生成の流れ

7/21



Fortran95 Parser (オリジナル版)

9 / 21

■ Fortran90/95 の文法を解釈

- RDoc のタグは XML に比べてとても簡潔
- 別ファイル内のモジュール等へ自動的にリンク作成

Fortran95 ソースコード

```

|= Module module_name_mod : Sample module
| Authors:: Yasuhiro MORIKAWA
|
| This module depends base_mod module
|
module sample_mod
  use base_mod
  implicit none
  private
  public :: sample_init, sample_end, Const
  real(8) :: Const = 3.14

  subroutine sample_init(inchar, outint)
    character(*) , intent(in) :: inchar
    integer(INTKIND), intent(out):: outint
  end subroutine sample_init

  subroutine sample_end(err)
    logical, intent(inout) :: err
  end subroutine sample_end
end module sample_mod
    
```

Rdoc

HTML のリファレンスマニュアル

Files	Classes	Methods
sample.f90 base.f90	sample_mod base_mod	sample_init sample_end
Class <code>sample_mod</code>		
In: <code>sample.f90</code>		
Module <code>sample_mod</code> : Sample module Authors: Yasuhiro MORIKAWA This module depends base_mod module		
Methods		
sample_init sample_end		
Included Modules		
base_mod		
Public instance methods		
<code>sample_init</code> (<i>inchar, outint</i>)		
[source]		
<code>sample_end</code> (<i>err</i>)		
[source]		

ハイパーリンク

Fortran95 Parser の問題点

10/21

■ 解析機能の不足

The screenshot shows a web-based interface for a Fortran95 parser. At the top, there are three tabs: 'Files', 'Classes', and 'Methods'. Under 'Files', 'base.f90' and 'sample.f90' are listed. Under 'Classes', 'base_mod' and 'sample_mod' are listed. Under 'Methods', 'base_init (base_mod)' and 'sample_init (sample mod)' are listed. The main content area displays the details for the 'sample_mod' class, including its location ('In: sample.f90') and authors ('Yasuhiro MORIKAWA'). Below this, the 'Public Instance Methods' section is visible, showing the signature 'sample_init(char, outint)'. An orange callout box with white text points to this signature, stating 'サブルーチンの引数の型を表示できない' (Cannot display the type of subroutine arguments).

Fortran95 Parser の問題点

11 / 21

■ 解析機能の不足

The screenshot shows a graphical user interface for a Fortran95 parser. At the top, there are three tabs: 'Files', 'Classes', and 'Methods'. Under 'Files', 'base.f90' and 'sample.f90' are listed. Under 'Classes', 'base_mod' and 'sample_mod' are listed. Under 'Methods', 'base_init (base_mod)' and 'sample_init (sample_mod)' are listed. The main area displays details for the 'sample_mod' class, including its location ('In: sample.f90'), its name ('Module sample_mod : Sample module'), and its author ('Authors: Yasuhiro MORIKAWA'). Below this, there are sections for 'Methods' (listing 'sample_init'), 'Included Modules' (listing 'base_mod'), and 'Public Instance method' (listing 'sample_init' with parameters 'inchar, outint'). A blue arrow points from an orange callout box to the 'sample_init' method entry in the 'Public Instance method' section.

Files
base.f90
sample.f90

Classes
base_mod
sample_mod

Methods
base_init (base_mod)
sample_init (sample_mod)

Class **sample_mod**
In: sample.f90

Module sample_mod : Sample module
Authors: Yasuhiro MORIKAWA

Methods
sample_init

Included Modules
base_mod

Public Instance method
sample_init *inchar, outint*
[Source]

サブルーチンの
解説コメントを表示
できない

Fortran95 Parser の問題点

12/21

■ 解析機能の不足

The screenshot shows a graphical user interface for a Fortran95 parser. It has three tabs at the top: 'Files', 'Classes', and 'Methods'. Under 'Files', 'base.f90' and 'sample.f90' are listed. Under 'Classes', 'base_mod' and 'sample_mod' are listed. Under 'Methods', 'base_init (base_mod)' and 'sample_init (sample mod)' are listed. The 'sample_mod' class is selected, showing 'In: sample.f90'. Below this, it says 'Module sample_mod : Sample module' and 'Authors: Yasuhiro MORIKAWA'. There is a 'Methods' section with 'sample_init'. An 'Included Modules' section lists 'base_mod'. A 'Public Instance methods' section is underlined in red and contains 'sample_init' with parameters 'inchar, outint'. An orange callout box with an arrow pointing to this section contains the text: 'public, private を 区別できない (全て public 扱いになる)'. The bottom of the window shows a taskbar with 'インターネット'.

Fortran95 Parser の問題点

13 / 21

■ 解析機能の不足

The screenshot shows a graphical user interface for a Fortran95 parser. At the top, there are three panes: 'Files' (listing base.f90 and sample.f90), 'Classes' (listing base_mod and sample_mod), and 'Methods' (listing base_init (base_mod) and sample_init (sample_mod)). The main area displays the details for the 'Class sample_mod' (In: sample.f90). It includes sections for 'Module sa...', 'Methods' (with sample_init), 'Included Modules' (with base_mod), and 'Public Instance me...'. An orange callout box with a downward-pointing arrow is overlaid on the interface, containing the text 'そもそも表現できない要素いろいろ' and a list of features not supported by the parser.

Class **sample_mod**
In: sample.f90

Module sa
Authors: Yasuh...

Methods
sample_init

Included Modules
base_mod

Public Instance me
sample_initinchar, ou
[Source]

インターネット

そもそも表現できない要素いろいろ

- ◆ 関数 (function 文)
- ◆ モジュールが公開する変数, 定数
- ◆ 構造体 (type 文)
- ◆ 利用者定義演算子 (operator)
- ◆ 利用者定義代入 (assignment)
- ◆ 総称手続き (interface 文)

etc ...

問題点の解決に向けて

14 / 21

■ 開発者・メンテナに連絡

- 森川 「こうなると嬉しいのだけど」と連絡
- 開発者 「メンテナさんに連絡しておいたよ」
- メンテナ 「やってみてくれます?」
- 森川 (T_T)

■ しょうがないので (?) 自力で改造

- 見た感じで足りないと思うところから五月雨的に解析機能追加

Fortran95 Parser (強化版)

15 / 21

The screenshot shows a web browser window displaying the Fortran95 Parser (強化版) interface. The address bar shows the URL: http://shadow.vir/morikawa/rdoc_sample/doc/. The interface is divided into several sections:

- Files:** base.f90, sample.f90
- Classes:** base_mod, sample_mod
- Methods:** TYPE_A (sample_mod), base_init (base_mod), const (sample_mod)

The main content area displays the details for the **Class sample_mod**, with **In: sample.f90**. Below this, there is a section for **Derived Types** showing **TYPE_A**. Underneath, the **Public Instance Methods** section lists **TYPE_A()** and **Derived Type :** with a variable **counter : integer** and its explanation: **: 構造体内部の変数の解説**. At the bottom, there is a section for **構造体の解説**.

構造体の解析
要素、要素の型、解説の表示

Fortran95 Parser (強化版)

16 / 21

```
const()
Constant :
const = 3.14 : real(8), parameter
           : 公開定数
```

```
sample_fun
Function :
```

定数の解析
型、初期値、解説の表示

[\[Source\]](#)

```
sample_init( inchar, outint )
Subroutine :
inchar : character(*) , intent(in)
           : 入力変数
outint : integer(INTEGER), intent(out)
           : 出力変数
```

初期化サブルーチン

[\[Source\]](#)

Fortran95 Parser (強化版)


17 / 21

```
const()
Constant :
const = 3.14 : real(8), parameter
           : 公開定数
```

```
sample_func( log ) result(res)
Function :
res :      logical
       : 論理型の返り値
log :     logical, intent(in)
       : 論理型入力変数
```

関数

[\[Source\]](#)



関数の解析
解説の表示

初期化サブルーチン

[\[Source\]](#)

Fortran95 Parser (強化版)

18 / 21

```
const()
Constant :
const = 3.14 : real(8), parameter
           : 公開定数
```

```
sample_func( log ) result(res)
Function :
res :      logical
           : 論理型の返り値

log :      logical, intent(in)
           : 論理型入力変数
```

関数

[Source]

```
sample_init( inchar, outint )
Subroutine :
inchar :    character(*), intent(in)
             : 入力変数

outint :    integer(INTEGER_KIND), intent(out)
             : 出力変数
```

初期化サブルーチン

[Source]

引数の解析
型、解説の表示

Fortran95 Parser (強化版)

19 / 21

関数

[Source]

```
sample_init( inchar, outint )
```

Subroutine :

```
inchar : character(*), intent(in)  
        : 入力変数
```

```
outint : integer(INTKIND), intent(out)  
        : 出力変数
```

初期化サブルーチン

[Source]

Private Instance methods

```
internal()
```

Variable :

```
internal : integer, save  
          : 非公開変数
```

[Validate]

インターネット

public, private
を区別

Fortran95 Parser (強化版)

20 / 21

閉鎖

[Source]

```
sample_init( inchar, outint )
```

Subroutine :

```
inchar : character(*), intent(in)  
       : 入力変数
```

変数の解析
型、初期値、解説の表示

Private Instance methods

```
internal()
```

Variable :

```
internal : integer, save  
         : 非公開変数
```

[Validate]

インターネット

RDoc-f95 の使用例: spmodel

現在の開発/メンテ状況

- メンテナ交代
 - 2011 年より佐々木が開発/メンテすることに.
 - 最後の更新が 2009 年なので, そろそろ更新予定
- 最近の更新/追加機能
 - RDoc 3.x への対応
 - Parser
 - 日英併記の要望がある?
 - fortran2003 への対応?
 - Generator
 - Syntax highlighter の追加
 - roff(man), HTML5 (作成中)

- 1 話の「まくら」
- 2 リファレンスマニュアルの作成アレコレ
- 3 RDoc-f95
- 4 今後の展望/議論の種

- RDoc
 - 2004 年まで Ruby 本体で開発. 2008 年から Ruby 本体とは別に開発
 - 2010 年から RDoc 3.x の開発が開始
 - 問題点: Ruby1.9 対応, シンプル過ぎるタグ, Ruby 同梱版との不整合
- YARD: Yay! A Ruby Documentation Tool
 - 2007 年から開発開始
 - RDoc との完全互換
 - Doxygen, Javadoc の様な「タグ」をサポート
 - 複数のマークアップを混在可能
 - 長い文章は $\text{T}_\text{E}\text{X}$, RD, Markdown で書く, など

...

おしまい

参考文献/URL

- Kernighan, B.W., Plauger, B.J., 木村 泉 (訳), 1982: 「プログラミング書法」, 第 2 版, 共立出版, ISBN 978-4320020856
- 森川靖大, 石渡正樹, 堀之内武, 小高正嗣, 林祥介, 「RDoc を用いた Fortran90/95 プログラムのドキュメント生成」, 地球惑星科学連合 2006 年大会, 幕張メッセ国際会議場, 2006 年 5 月 14 日 (講演番号 J157-016)
- Doxygen: <http://www.stack.nl/~dimitri/doxygen/>
- Javadoc:
<http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html>
- GTK-Doc: <http://www.gtk.org/gtk-doc/>
- RDoc: <http://rubyforge.org/projects/rdoc/>
- Sphinx: <http://sphinx.pocoo.org/>
- BitClust: <http://doc.loveruby.net/wiki/BitClust.html>
- Texinfo: <http://www.gnu.org/s/texinfo/>
- RDTool: <http://raa.ruby-lang.org/project/rdtool/>