504：*4011*（RDoc；Fortran；ドキュメント生成；数値モデル；Ruby）

RDocを用いた数値モデルのドキュメント生成*

森川靖大[*1]・石渡正樹[*2]・堀之内武[*3]
小高正嗣[*4]・林祥介[*4]

1．はじめに

この小論では，Fortran 90/95で書かれた数値モデルに付随するドキュメント生成とその管理の為になされてきた試みについて紹介し，オブジェクト指向スクリプト言語Rubyによるドキュメント生成用のライブラリRDocを用いた手法について解説を行なう．

数値モデルの開発や保守において，モデルを構成するサブルーチンや関数など，プログラム単位の機能や使用法の詳細を記述したドキュメント（リファレンスマニュアル）の整備を行うことは必須な作業である．数値モデルを利用する第三者にとっても，モデルを理解し，あるいはこれに改良を加える上で，このようなドキュメントが提供されていることは望ましい．

しかし，モデルの開発に合わせて常に最新のドキュメントを整備しておくことは大変に手間のかかる作業である．その主な原因は，モデルで記述した内容とほぼ同じ内容（サブルーチンの名前や引数など）を再度ドキュメントとしても記述せねばならない点にある．ドキュメントとモデルとを別々のファイルで管理する場合にはこの作業はさらに煩雑となる．そのためドキュメントの整備はついついおろそかになり，結果としてモデル本体の開発にも支障をきたすこととなる．

ドキュメント作成の手間を軽減し，ドキュメント整備を促進させる手法は，ソースコードのコメント行にドキュメント用の文書を埋め込むことによってモデルとドキュメントの一元管理を実現し，かつ，ドキュメント生成をソースコード解析用のソフトウェアによって自動化することである．例えば，まつもとゆきひろ氏を中心に開発されたオブジェクト指向スクリプト言語Ruby[†1]や，サン・マイクロシステムズ社により開発されたオブジェクト指向プログラミング言語Java[†2]では，それぞれRDoc[†3]，Javadoc[†4]といったドキュメント自動生成用のライブラリが用意されている．C，C＋＋，Python，IDLに関してはDoxygen[†5]というドキュメント生成用ソフトウェアが存在する．これらRDocやJavadoc，Doxygenは，単にソースコードの解析やコメント行中に記述される文書の抜粋を行なうだけではなく，自動生成されるHTMLドキュメントにハイパーリンクや見出し，箇条書きと

---

* Document generation for numerical models by RDoc.

[*1] Yasuhiro MORIKAWA, 北海道大学理学院. morikawa@gfd-dennou.org
[*2] Masaki ISHIWATARI, 北海道大学地球環境科学研究院.
[*3] Takeshi HORINOUCHI, 京都大学生存圏研究所.
[*4] Masatsugu ODAKA, Yoshi-Yuki HAYASHI, 北海道大学理学院.

[†1]「オブジェクト指向スクリプト言語Ruby」http://www.ruby-lang.org/．近年，Rubyは地球科学におけるデータ解析，可視化，数値シミュレーションに対しても活用されている．その例については，「電脳Rubyプロジェクト」http://ruby.gfd-dennou.org/を参照されたい．＜URL参照日付2007/01/18＞

[†2]「サン・マイクロシステムズ-Javaテクノロジ」http://jp.sun.com/java/＜参照2007/01/18＞

[†3]「RDoc：Ruby Standard Library Documentation」http://www.ruby-doc.org/stdlib/libdoc/rdoc/rdoc/＜参照2007/01/18＞

[†4]「Javadoc Tool Home Page」http://java.sun.com/j2se/javadoc/＜参照2007/01/18＞

[†5]「Doxygen」http://www.doxygen.jp/＜参照2007/01/18＞

いった構造を付与することで，見やすく使いやすいドキュメントの提供を可能としている．また，ソースコードの可読性を低下させないよう，文書に構造を付与するためのタグも簡素なものとなっている．上記以外にも同種のライブラリやソフトウェアは数多く存在しており，現在この手法はリファレンスマニュアル管理の主流となりつつある．

これまでに気象学分野の数値モデルに用いられてきた主要な言語はFortran系の言語であり，近年はFortran 90/95が主流である．Rubyによるドキュメント生成ライブラリRDocは，Rubyのみならず CやFortran 90/95のソースコードを解析してドキュメントを生成することも可能である．しかし残念ながらRDocのFortran 90/95解析機能には不十分な点が多く，実用には耐えなかった．そこで我々はこの不十分な点を解決するべく，Fortran 90/95解析機能の改良，強化を行なった．以下では，RDoc以外のFortran 90/95プログラムのドキュメント生成方法について紹介した後に，RDocの特徴，および我々が施した改良点について解説を行う．そして，この「RDoc Fortran 90/95ソースコード解析機能強化版」（以下，RDoc強化版）を用いたドキュメント生成の方法を実例を交えて紹介する．ここで紹介するRDoc強化版はインターネット上に公開されており，現在その改良点がオリジナルのRDocライブラリ[†3]に取り込まれつつある．詳細な説明，最新の状況については当該ホームページ[†6]をご覧いただきたい．

2．Fortran 90/95プログラムのドキュメント生成

RDoc以外にも，Fortran 90/95プログラムのドキュメント生成に関する試みはなされている．ここでは，f90docとf90tohtmlというソフトウェアおよびGFDLにおける取り組みについて紹介しておく．

f90doc[†7]はMITの計算科学グループのErik Demaine氏によって作成されたFortran 90/95用のドキュメント自動生成Perlスクリプトである．このソフトウェアはFortran 90/95ソースコードを解析して，サブルーチンや関数の直前に記述されるコメントからHTML形式のドキュメントを生成する．さらに，参照するモジュールへのハイパーリンクの作成，サブルーチンや関数の引数のデータ型，INTENT属性（授受特性），引数キーワード（例：“call date_and_time（values＝input）”における“values”）も自動的にドキュメントに反映する．しかし，コメントに記述される内容はHTMLへ変換されても，ハイパーリンクや見出しや箇条書きといった構造を持たない文書となるため，リファレンスマニュアルとしては不便である．他にも，Fortran 90/95の新機能である多重定義を使用したプログラムが正しく解析されないなどの問題がある．

f90tohtml[†8]は，オクラホマ大学気象学講座のBrian Fiedler氏ほかによって作成された，Fortran 90/95ソースコードをHTMLへ変換するPerlスクリプトである．このソフトウェアは大規模な数値モデルのソースコードを閲覧することを目的に開発され，現在はARPS，WRF，CAMにおいて利用されている．f90tohtmlは，USE文中のモジュール名やCALL文中のサブルーチン名を，該当するモジュールまたはサブルーチンのソースコードへのハイパーリンクに変換する．利用者はソースコードをブラウザで閲覧でき，プログラムが参照している別のモジュールやサブルーチンへはハイパーリンクによって移動できる．しかしながら，このソフトウェアはサブルーチンや関数などの機能や使用法を詳述したリファレンスマニュアルの作成を主目的とはしていないので，ソースコード内のコメント行を自動解析してこれを取り込み，その記述をハイパーリンクや見出しや箇条書きなどの構造を持ったHTML文書として生成する機能は有していない．また，PRIVATE文によるモジュール中の言語要素への参照不許可の指定や，構造データ型，多重定義などを用いたプログラムを正しく解析できないという問題もある．

GFDLにおいてIsacc Held氏とVenkatramani Balaji氏が中心となって開発を行なっている気候モデルの統合フレームワークFlexible Modeling System（FMS）[†9]では，ソースコード内のコメント行にXML書式のドキュメントを埋め込むことで，ドキュメントの作成と管理の手間を軽減している．HTML形式のドキュメントを生成する場合には，ソースコー

[†6]「RDoc Fortran 90/95ソースコード解析機能強化版」http://www.gfd-dennou.org/library/dcmodel/rdoc-f95/＜参照2007/01/18＞

[†7]「f90doc homepage」http://theory.lcs.mit.edu/~edemaine/f90doc/＜参照2007/01/18＞

[†8]「f90tohtml Homepage」http://mensch.org/f90tohtml/＜参照2007/01/18＞

[†9]「FMS：the flexible modeling system」http://www.gfdl.noaa.gov/~fms/＜参照2007/01/18＞

ドからXML部分を抽出し，変換を行うようにしている．この手法の詳細はFMSプロジェクトのホームページ上に公開されている[†10]．FMSで用いられている手法では，ドキュメントをXMLで記述するため，ドキュメントにハイパーリンクや箇条書きなどの構造を持たせることが可能であるという利点がある．しかし，コメント行にXMLタグを数多く記述する必要があり，Fortran 90/95プログラムとしてのソースコードが読みにくくなるという欠点がある．また現時点では，ソースコードからXML部分を抜き出し，HTML形式のドキュメントを生成するソフトウェアは公開されておらず，この手法を誰でもがすぐに利用できるようにはなっていない．

このように，気象学分野のモデルを含めたFortran 90/95プログラムにおいても，他の言語と同様にドキュメント作成と管理のための試みはなされつつある．しかし，RubyやJavaなどの言語のように，プログラムとドキュメントを1つのファイルとして管理でき，かつプログラム中のコメント行に簡素なタグを付加することによってハイパーリンクや箇条書きなどの構造を持つHTMLドキュメントを自動生成できるような手法は未だに確立されておらず，モデルのドキュメント整備のための手間の軽減は思うようには進んでいない．

3．RDocの特徴

RDocはプログラムを解析し，ソースコードから得られる情報と，コメント行に記述された文書からHTML形式（正確にはXHTML 1.0 Transitional形式）のドキュメントを生成するライブラリである．RDocが主に対象としているのはもちろんRubyスクリプトであり，Rubyの標準ライブラリのドキュメントの多くはRDocによって作成されている．

RDocの特徴として，以下の3点が挙げられる．1点目は，Rubyのソースコードを解析することによって，クラス名やメソッド名，クラス間の継承の関係などを自動的にドキュメントに反映することである．この機能により，ソースコードに記述する内容をドキュメントに再度記述する必要が無くなり，ソースコードの内容に沿った最新のドキュメントを整備することが簡単になる．2点目の特徴は，コメントとして記述する文書に簡素なタグを付加することで，箇条書きやハイパーリンク等の文章構造をHTMLドキュメントに付与できることである．この機能により，ソースコードの可読性をさほど犠牲にすること無く，使い勝手の良いドキュメントの作成が可能となる．3点目の特徴はクロスリファレンスと呼ばれる機能を持つことである．これは，複数のプログラムからドキュメントを生成する際に，コメント行に記述されたクラス名やメソッド名を自動的にそのクラスやメソッドのドキュメントへのハイパーリンクへと変換する機能である．本来，ハイパーリンクを作成する際には，HTMLとして表示する文字列およびリンク先のURL，そしてハイパーリンクであることを示すタグを記述する必要がある．クロスリファレンス機能により，これらの文字列とタグを記入すること無く，コメント行中にクラス名やメソッド名を書いておくだけでハイパーリンクが自動作成される．これによって，コメントを記述する手間を減らすことができるだけでなく，ソースコードの可読性の低下を防ぐこともできる．

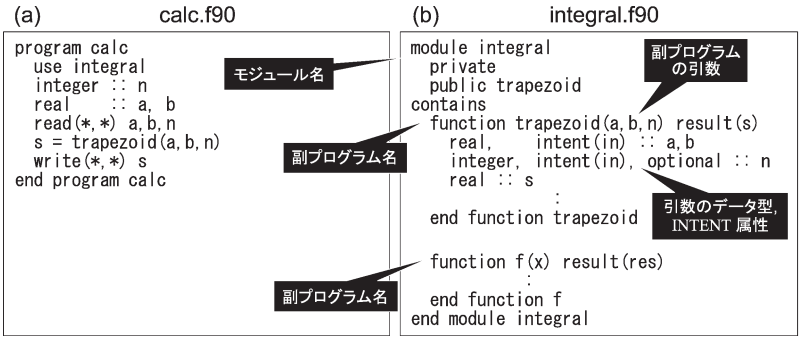
RDocはRubyのみならず，Fortran 90/95ソースコードの解析機能も有している．しかし，残念ながら既存のRDocユーティリティーでは，Fortran 90/95プログラム中のファイルやモジュールやサブルーチン以外のプログラム要素，すなわち関数や変数などを解析することができなかった．またFortran 90/95プログラムのドキュメントとしては必須である，サブルーチンの引数の情報，すなわちデータ型，INTENT属性（授受特性），引数キーワードなどが解析されなかった．そしてコメントを解釈してドキュメントに取り込む機能も正しく働かないという問題があった．

4．RDoc Fortran 90/95解析機能の強化

我々は，前節で述べた問題点を解決するとともに，より充実した情報を持つFortran 90/95プログラムのリファレンスマニュアル自動生成が可能となるように，RDocのFortran 90/95ソースコード解析機能の改良，強化を行った．我々が改良を施したRDocは，Fortran 90/95ソースコードのほとんどの言語要素，すなわち主プログラム，モジュール，サブルーチン，関数，変数，定数，構造データ型，利用者定義演算子，利用者定義代入，NAMELIST文を解析することができる．それらの言語要素の直後（行末もしくは直後に続く行）に書かれたコメント行は，その言語要

[†10]「The FMS Manual A developer's guide to the GFDL Flexible Modeling System」http://www.gfdl.noaa.gov/~vb/FMSManual/＜参照2007/01/18＞

素の解説文章として識別される．この他にもモジュール間の依存関係，総称名を用いた多重定義，PRIVATE文によるモジュール中の言語要素の参照不許可の指定なども解析可能である．サブルーチンや関数に関しては，その引数のデータ型，INTENT属性(授受特性)，引数キーワードをソースコードから自動的に解析すると共に，引数の宣言文の直後に書かれたコメントをその引数に関するドキュメントとして識別する．



第1図　例として用いるFortran 90/95プログラム．(a) は主プログラム，(b) は主プログラムから参照されるモジュールを記述したファイルである．

Fortran 90/95のソースコード解析機能以外にもいくつかの改良を行なっている．オリジナルのRDocでは実現できていない，ファイル名のクロスリファレンス機能を追加した．また，北海道大学数学専攻の黒田　拓氏によるRuby用MathMLライブラリ[†11]を組み込む機能をRDocに付加することで，TeX形式の数式をコメントに埋め込んだ場合，その数式をMathML形式に変換してドキュメントに反映するようにした．

以上の試みは現在RDocのオリジナルライブラリ[†3]に順次組み込まれつつある．詳細は我々の公開ホームページ[†6]を参照していただくことにし，次節で具体例を簡単に紹介することにする．

5．RDocによる数値モデルのドキュメント生成

ここではRDoc強化版を用いたドキュメントの生成方法を，実例を挙げて紹介する．RDoc強化版は，Rubyがインストールされている計算機環境なら容易に導入することができる．具体的なインストール方法はRDoc強化版ホームページ[†6]をご覧いただきたい．

(1) Fortran 90/95プログラムの作成

例として，第1図に示す簡単なFortran 90/95プログラムを考える．(a) は主プログラムcalcが記述されたファイル，(b) は主プログラムから参照されるモジュールintegralが記述されたファイルである．

(2) ドキュメントの生成

ドキュメント生成はコマンドライン上から行う．プログラムの置いてあるディレクトリへ移動した後，以下のようにコマンド入力する．

```
% rdoc --ignore-case --inline-source
```

これによりdoc/というディレクトリが作成される．doc/index.htmlをブラウザで閲覧すると，第2図に示されるようなドキュメントが表示される．上部のフレームにはファイル（Filesの欄），モジュール(Classesの欄)，サブルーチン，関数など（Methodsの欄）のリストが表示される．下部のフレームにはサブルーチンや関数に関する情報が記載された，簡単なリファレンスマニュアルが表示される．

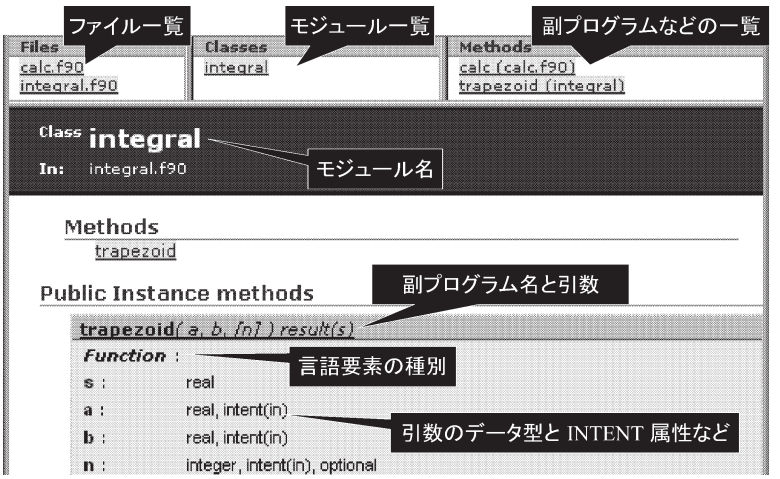
(3) ドキュメント用文書の埋め込み

上記のプログラムに実際にドキュメント用文書を埋め込んでみる．基本的にはサブルーチンや関数など，言語要素の宣言文の行末もしくは直後の行にコメントを記載する．HTMLドキュメントに箇条書きやハイパーリンクなどの構造を持たせたい場合は，以下のタグを用いる．

見出し：行頭に“＝”，“＝＝”，“＝＝＝”，“＝＝＝＝”を記述する．“＝”の数が見出しのレベルに対応する．

箇条書き：行頭に“＊”，または“-”を記述する．番号付き箇条書きは“1．”，“2．”，...，アルファベット付き箇条書きは“A．”，“B．”，...，を記述する．入れ子にする場合，半角2文字分字下げして再度行頭に上記の文字

[†11]「ひらくの工房-MathMLライブラリ」http://www.hinet.mydns.jp/~hiraku/. MathMLとはXMLベースの数学用マークアップ言語であり，数式などの数学的情報もHTMLのようにWeb上で公開，受信，処理することを目的としたものである．詳しくは「W3C Math Home」http://www.w3.org/Math/を参照されたい．＜参照2007/01/18＞



第 2 図　RDoc を用いて作成した，第 1 図のプログラム群のドキュメント．

integral.f90

```
module integral
        :
contains
  function trapezoid(a,b,n) result(s)
    !=== 台形公式を用いた数値積分のプログラム例
    !
    ! $ f(x) = x^2 - 2x + 2 $ の積分値を台形公式によって求める.
    !
    ! * 積分区間は $ [a, b] $ とする.
    ! * 実際に計算する式は以下の通り.
    !     ここで $ h $ は $ h = ¥frac{b-a}{n} $ である.
    !
    ! ¥[
    !   ¥int_{a}^{b} f(x) dx =
    !     ¥frac{h}{2} ¥left¥{f(a) + f(b)¥right¥} + h¥sum_{i=1}^{n-1} f(a+ih)
    ! ¥]
    !
    ! このサンプルプログラムは http://www.gfd-dennou.org/library/dcmodel
    ! にて入手できます.
    !
    real,    intent(in) :: a ! 積分区間の下限
    real,    intent(in) :: b ! 積分区間の上限
    integer, intent(in), optional :: n ! 積分区間の分割数
    real :: s  ! 積分値
        :
  end function trapezoid
        :
end module integral
```


第 3 図　第 1 図 b のソースコードのコメント行にドキュメント用文書を埋め込んだもの．

を記述する．

ラベル付き箇条書き：ラベルを角括弧で括る（“[label] contents”），またはラベルと内容を“：：”で区切る（“label：：contents”）．

ハイパーリンク：表示する文字をリンク先の URL と同じにする場合は，“http://”などで始まる URL をそのまま記述する．表示する文字とリンク先の URL を別にする場合は，“{displayed message} [URL]”と記述する．ファイルやクラスやメソッドのドキュメントへのリンクは，ファイル名やクラス名やメソッド名をそのまま記述する．

数式の MathML 表示：TeX 書式の数式を＄～＄もしくは¥［～¥］で括る．

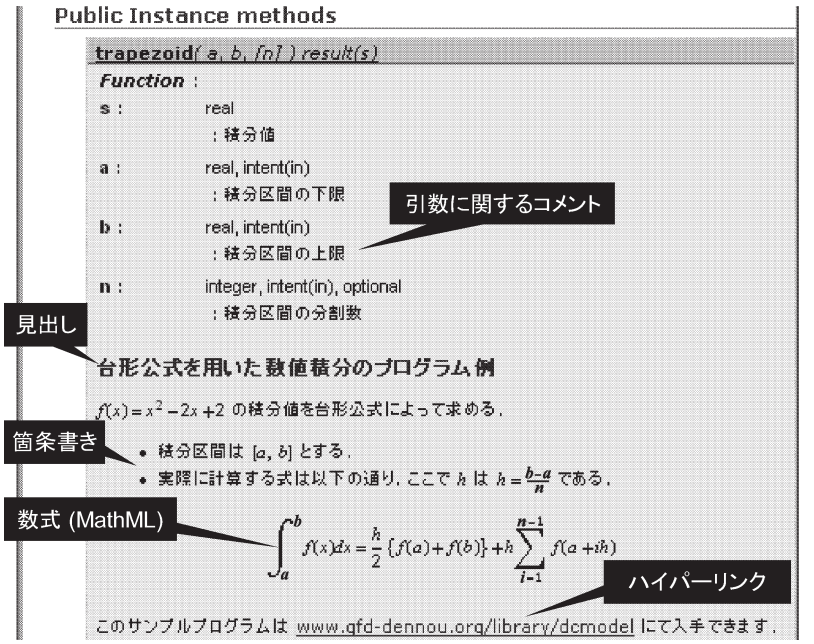
文字の修飾：文字を斜体で表示する場合は，“_word_”もしくは“＜em＞word＜/em＞”，太字で表示する場合は，“＊word＊”もしくは“＜b＞word＜/b＞”，等幅で表示する場合は“＋word＋”もしくは“＜tt＞word＜/tt＞”と記述する．

第 1 図（b）のソースコードにドキュメント用文書を埋め込んだ具体例を第 3 図に示す．

これらのプログラムから，再度 RDoc によってドキュメントを生成する．今回示す例のようにコメントに日本語を埋め込む場合には，ファイルの文字コードに応じた引数オプションを指定する．下記では“euc-jp”を指定している．また，TeX で記述した数式を MathML で表示するために“--mathml”オプションも使用する．

```
% rdoc --ignore-case --inline-source ¥
      --charset euc-jp --mathml
```

ここで“¥”は継続行を表す．doc/ディレクトリの内容が更新されるので，再度 doc/index.html をブラウザで閲覧する．今度は trapezoid 関数の部分に，第 4 図に示されるようなドキュメントが表示されるは



第 4 図　第 3 図に示した関数 trapezoid のドキュメント．

ずである．この図では，先ほどソースコードのコメントに埋め込んだ記述（“＝＝＝”や“＊”，URL，TeX の数式など）が HTML における見出し，箇条書きなどへと変換されている．なお，MathML を第 4 図のように表示するためには，ブラウザが MathML に対応している必要がある．対応ブラウザなどの情報に関しては RDoc 強化版ホームページ[16]を参照していただきたい．

6．おわりに

この小論で紹介した RDoc 強化版はプログラムの規模によらず簡単に使用することができる．また，プログラム中に全くコメントが無くても第 2 図に示す程度のドキュメントを生成することはできる．したがって，ドキュメント整備におけるプログラム開発者の手間の軽減だけでなく，ドキュメントの用意されていないプログラムの使用者がプログラムを理解するための助けになることも期待できる．我々は，このライブラリの利便性と完成度を高めるべく，今後も継続して改良を施していく予定である．読者からのコメントや意見，情報などがあれば，筆者ら開発グループ宛（E-mail：dcmodel-rqst@gfd-dennou.org）にお寄せいただきたい．

このライブラリが，数値モデルに携わる方々の手助けになれば幸いである．