

Gfdnavi における RESTful な Web サービスの提供

諫本 有加[†] 渡辺 知恵美[‡] 堀之内 武^{‡‡} 西澤 誠也^{‡‡‡}

[†] 奈良女子大学理学部情報科学科 〒630-8506 奈良県奈良市北魚屋西町

[‡] お茶の水女子大学理学部情報科学科 〒112-8610 東京都文京区大塚 2-1-1

^{‡‡} 北海道大学地球環境科学研究所 〒060-0810 北海道札幌市北区北 10 条西 5 丁目

^{‡‡‡} 神戸大学大学院理学研究科 〒657-8501 兵庫県神戸市灘区六甲台町 1-1

E-mail: [†] isa3@ics.nara-wu.ac.jp, [‡] chiemi@is.ocha.ac.jp,

あらまし 近年、種類やサイズ等が増加している科学データに対して検索・解析・可視化を簡単に行うことが出来るツール Gfdnavi の開発を進めている。Gfdnavi における検索・解析・可視化の作業は Web ブラウザでの GUI ベースで行うが、それらの作業をプログラム上で実行させたいという要求がある。そこで、我々はデータの検索・解析・可視化の一連の作業を行うことのできる REST-style の Web サービスを実装し、ユーザの作業目的に柔軟に対応できる Web サービスの提供を行った。また、その Web サービスを使い、複数のサーバに対する横断検索の開発を行った。

キーワード REST, Web サービス, 横断検索

Providing REST-style Web Services for Gfdnavi

Yuka ISAMOTO[†] Chiemi WATANABE[‡] Takeshi HORINOUCI^{‡‡}
and Seiya NISHIZAWA^{‡‡‡}

[†] Department of Information Sciences, Faculty of Science, Nara Women's University
Kitauoyanishi, Nara-shi, Nara, 630-8506 Japan

[‡] Department of Information Sciences, Faculty of Science, Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

^{‡‡} School of Environmental Earth Science, Hokkaido University
10-5, Kita-ku, Sapporo-shi, Hokkaido, 060-0810 Japan

^{‡‡‡} Department of Earth and Planetary Sciences, Graduate School of Science, Kobe University
1-1 Rokkodai, Nada-ku, Kobe-shi, Hyogo, 657-8501 Japan

E-mail: [†] isa3@ics.nara-wu.ac.jp, [‡] chiemi@is.ocha.ac.jp,

Abstract We are developing Gfdnavi, which is a integrated desktop tool and it can search, analyze and visualize geophysical fluid data. This tool also works as a web server, then the user can publish these data and the results of processes as web contents. In this paper we describe REST-style web services for Gfdnavi. Gfdnavi has provided the rich web interface for searching, analyzing, and visualizing geophysical fluid data. However we can consider the case that the users want to execute these processes from their programs. Then, we develop REST-style web services for data searching, analyzing, and visualizing geophysical fluid data on Gfdnavi. We also provide a cross-search mechanism among public Gfdnavi servers by using the web services.

Keyword REST, web services, Cross-Search

1. はじめに

近年、地球観測測定器の発達と計算機の高速化により、大気や海洋などの地球流体データが増加している。一般企業や NASA などは、データセンタを設置し膨大な地球流体データを管理して Web 上で公開しており、世界中の科学者が利用できるようになっている。科学

者達は自分の計算機に蓄積されたデータを検索するだけでなく、同様の研究を行う科学者同士で観測データを公開・共有し合いたいと要求している。しかし、一般の科学者が自分のデータを検索・公開するには作業コストや学習コストがかかる。こういった科学者の要求を満たすべく、地球流体データのデータアーカイブ

サーバ構築支援ツール：Gfdnaviを共同で開発している。著者らを含むグループは、これまで地球流体データのメタデータの管理・解析・可視化用のRubyパッケージとしてDCL、GPhysを開発し、提供してきた[4]。それらのパッケージをベースにデスクトップ上のデータの検索・解析・可視化の一連の流れおよびサイクルを扱う総合的デスクトップツールにしたのがGfdnaviである。さらに、解析・可視化によって得られた知見情報を整理して再利用することもできる。また、デスクトップツールの機能をWebコンテンツとして公開することもでき、公開するデータの限定やユーザ認証・アクセス制御も備えている。実際に、Gfdnaviを利用したサーバ[5]が一般公開されている。

Gfdnaviにおける検索・解析・可視化の作業はWebブラウザでのGUIベースで行う。しかし、それらの作業をプログラム上で実行させたいという要求がある。例えば、はじめにGUIを用いてどのようなデータがあるか、またデータにはどのような特徴があるかを確かめておき、それらのデータをどのように可視化すればよいか見当がついたら、その後にスクリプトを使って効率よく網羅的に可視化結果を得る場合である。そこで、我々はGfdnaviにおいてデータの検索・解析・可視化の一連の作業を行うことのできるREST-styleのWebサービスを提供することにした。これを提供することで、望むソースをURLの指定により取得できるようなる。したがって、上記に述べたような網羅的に可視化結果を得るスクリプトに対応することが可能となる。また、クライアントからGfdnaviのWebサービスにアクセスするためのパッケージを合わせて提供する。このライブラリはGPhys、DCLパッケージに基づいて作成するため、ユーザは従来組んできたスクリプトとほぼ同様の記述でリモートのGfdnaviサーバでのデータ検索・解析・可視化を行うことができる。また、このWebサービスを利用し、複数のGfdnaviサーバに対して横断的に検索を行う検索インタフェースを実装した。Gfdnaviでは科学データファイルに付与するメタデータの属性に対して属性名などの規定を行っていないため、公開者によって属性名が異なる場合があり、利用者は望むデータを検索するためのキーワードに何を指定してよいかわからない場合がある。そこで、Gfdnaviでは検索インタフェースにFaceted Navigationという対話的な絞り込み検索インタフェースを提供している[2]。この検索インタフェースを横断検索においても同様に利用できるよう実装を行った。

本稿では2節でGfdnaviについて、3節でWebサービスの目的について、4節ではWebサービスについて、5節で横断検索について述べる。6節でまとめを提示する。

2. Gfdnavi

Gfdnaviは、科学者が個人で持つ膨大な科学データをローカルで検索したり、共同研究者に対して公開・共有することを目的としたツールである。Gfdnaviは、ディスク上の科学データからメタデータを自動抽出して関係データベースとしてメタデータのデータベースを作成し、データの検索や検索した結果に対して解析・可視化を行う。Gfdnaviでは、地球流体の多次元数値データを扱うためのライブラリにGPhysを提供しており、数値データを「離散化された連続空間の物理量」として抽象化し、座標に関するメタデータ、単位・物理量の名前等に関するメタデータ、任意の名前と値の組で表されるメタデータを持つ。数値データの構成を図1に示す。

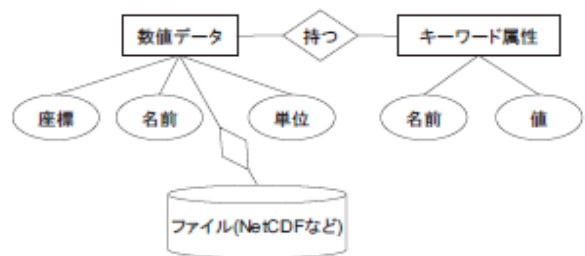


図1：数値データの構成

Gfdnaviの画面を図2に示し、特徴を以下に示す。

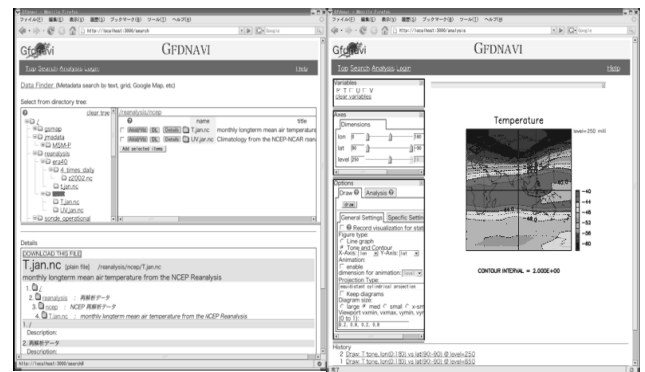


図2：ディレクトリ検索画面（左）

Gfdnaviの可視化画面（右）と

- 対象データを取めるディレクトリツリーをスキャンして自動的にメタデータを抽出して、DB化する。
- 科学データの検索・解析・可視化を行うことができる。
- 解析・可視化結果のダウンロードやアカウントがあればそれらをサーバに保存することができる。
- デスクトップアプリケーションとして個人利用することができるだけでなく、Webサーバ上でWebサイトとして利用することができる。

Gfdnaviでは、データのNetCDFファイルをダウンロードすることができる。NetCDFとは、大気・海洋科学の分野で広く使われるデータフォーマットで、変数に付属すべき情報（例えば名前、単位、配列の大きさなど）やファイルそのものに関する情報（例えばデータの製作者や製作日）が記述してある。したがって図2の左図のように、データのあるサーバ上で可視化を行うことができるだけでなく、そのサーバからNetCDFファイルをダウンロードして、自分のコンピュータ上で可視化を行うこともできる。NetCDFファイルのダウンロードについては4.5節で述べる。

さらに、現段階のGfdnaviは、Gfdnavi内で作成された図に対し、そこから得られた知見についての文章を作成し、図とともにドキュメントとして保存できるようになっている[6]。文章化された知見と図を合わせたものを知見情報と呼ぶ。知見情報はGfdnavi内で作成し、データベースに数値データや図、解析プロセスとともに知見情報を格納できるようになっている。これにより、数値データと知見情報とを関連付けて相互に検索を行えるようになっている。したがって、数値データを元に図や知見情報を検索することができる。例えば、ある研究者が気温データを解析してその図を描き、その説明となる文章を書いて知見文書として保存したとする。その気温データについて研究したい他の科学者はその図や文書の存在や置き場所を知らなくても、検索によって数値データを元にその知見文書を見つけ出すことができる。また、逆に図や知見情報からその元となった数値データや解析のプロセスを検索することもできる。例えば、気温について書かれた知見文書に対して、関連したデータと解析のプロセスを見つけることができるので、それを利用して解析を再現したり、また同じ解析手法を他のデータに適用したりすることもできる。図3に知見データの作成フォームを示す。知見についてのさらに詳しい説明は、[6]に書かれている。

3. Gfdnavi の Web サービスを提供する目的

Gfdnaviでは、科学データの検索・解析・可視化の作業はWebブラウザ上のGUIで行うことができるので、G-UIで繰り返し検索を行い、可視化した結果を見ながら欲しいデータに近づけていくことができる。しかし、G-UIで定型処理を繰り返し行うのは不適当であるため、はじめはこのGUIを使ってデータの種類や特徴などを確かめておき、それらデータの可視化の方法を知った上で、可視化結果を網羅的に取得するようなスクリプトを生成することが必要となる。例えば、日本付近の台風に興味を持っているユーザがいたとする。降雨観測のデータには、TRMMという衛星によりレーダーお

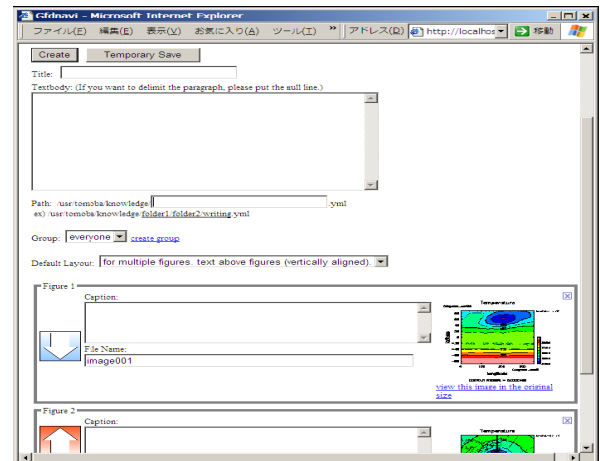


図3：知見情報作成フォーム

よびマイクロ波放射計で測ったデータがあるとする。まず、ユーザはこれらのデータに対してGUI上で可視化を行い、データの分解能やどのくらいの幅で降雨データを取得しているかの見当をつけた後、空間検索で日本付近に絞り検索を行う。そして、そのデータの中から台風に関するデータを取得するために、実際に台風が通った時間で絞り込みを行ったり、とにかく大量に可視化を行って台風のスパイラル構造が見えたら詳しく調べるといったことをするだろう。これらをGUI上で繰り返し行うのは手間がかかる。したがって、ユーザはこれらの処理を行うスクリプトを独自に作成するだろう。前者であれば、台風の中心の「ベスト」と思われる推定位置の軌跡（時間、経度、緯度の列）から時空間条件を緩めに抽出し、その上でマッチした全周回について網羅的に絵を描くプログラムを作成する。また、GUI上で可視化した図から利用できると思われるパラメタを見つけ、他のデータにも適用しようと考えた場合、そのパラメタを用いて可視化を行うプログラムを独自に作成するだろう。したがって、このような場合に対応できるようなWebサービスを提供することが本研究の目的である。そこで本研究では、WebサービスをRESTの原則に従わせることで、ユーザの作業目的に対応することができるシステムを開発した。RESTでは、全てのリソースはURLで表される一意的なアドレスを持っている。この原則に従い、検索結果を「リソース」として位置づけてURLを与えることで、そのURLにアクセスするだけで望む検索結果をデータとして取得できる。検索結果だけでなく、解析結果や可視化データもリソースとして位置づけURLを与えることができる。したがって、解析・可視化スクリプトはユーザが独自に作ったものを使用する場合は、データをユーザのコンピュータにダウンロードしたり、またデータが非常に大きかった場合には、解析・可視化

処理はそのサーバ上で行い、その結果のみをユーザのコンピュータにダウンロードしたりするといった選択ができる。このように検索・解析・可視化のステップをどこまでサーバ上で行うかの指定をユーザが自由に行えるなど、ユーザの作業目的に柔軟に対応できるシステムの提供が可能となる。

4. 提供する Web サービス

4.1. 提供する Web サービスの仕組み

3章より、提供するWebサービスの構成を図4に示す。

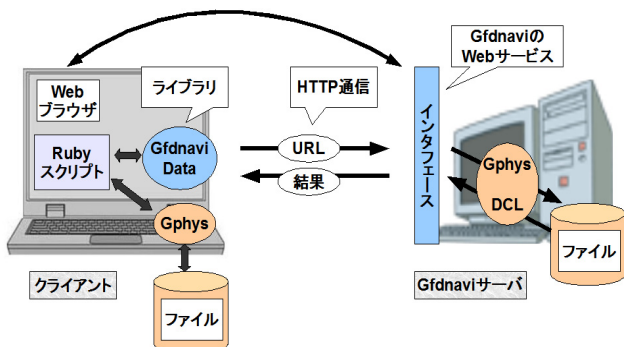


図4：提供するWebサービスの構成

ここでは特に GPhys と Gfdnavi のデスクトップ統合環境としての利用、Webサーバとしての利用、およびこのたび提供する Web サービスとの関係を示している。堀之内らはこれまで地球流体科学研究のための Ruby パッケージを提供しており、Gfdnavi はそれらのパッケージを GUI ベースで利用する、または Web ブラウザ・Webサーバを通してリモートサイトのデータの解析・可視化を GUI ベースで利用するためのツールという側面もある。

この度 Web サービスを提供することにより、ユーザは前節に述べたようなリモートサイトのデータの解析・可視化をプログラム上から行うことができるようになる。

ユーザは従来使ってきたパッケージの延長線上として Gfdnavi を利用することが好ましい。そこで我々は Ruby パッケージに Gfdnavi リモートアクセス用のクラス GfdnaviData, GfdnaviArray を提供する。

これらのクラスのメソッドは GPhys のデータアクセスメソッドにあわせて提供するため、ユーザはリモートにアクセスすることをほぼ意識しないでスクリプトを記述することができる。実際にはこれらのクラスは Gfdnavi サーバの Web サービスを実行し、その結果を受け取り、Ruby スクリプト上で扱う形に直してユーザに提示する。

ユーザが作成すると思われるスクリプトを図5に示す。

```
1: require "numru/gfdnavi_data"
2: include NumRu

3: root = GfdnaviData.parse("http://host:port")
4: samples=root.find(:all,:conditions=>"path.data/samples")
5: results=samples.find({:all,:conditions=>"kw.long_name=temperature
   and sp.overlap(0,0,90,90)"}))
6: DCL.groprn(1)
7: GGraph.set_fig("itr"=>2)
8: results.each{|t|
9:   t_cut=t.analysis("cut","lon=130")
10:  GGraph.tone(t_cut.to_gphys)
11:  GGraph.color_bar
12: }
13: DCL.grcls
```

図5：サンプルスクリプト

このサンプルスクリプトでは `http://host:port` 上にある Gfdnaviサーバに対して条件「パスが `data/samples` でありキーワード属性 `kw.long_name` の値が `temperature` である」を満たすデータを検索し、検索結果として得られる各データを経度130度で切り取り、処理を行ったデータをクライアントに持ってきて可視化する、という処理を行っている。ここではリモートのサーバに対して検索および解析処理を行い、そしてその結果をクライアントに持ってきてクライアントの可視化パッケージで処理している。このようなサーバとクライアントとのシームレスな関係を容易に記述することができる。

このサンプルスクリプトで GfdnaviData を利用しており、この中では 検索を行う `find` メソッド、解析処理を行う `analysis` メソッド、サーバ上のデータを取得し、クライアントで処理できる GPhys クラスのインスタンスに置き換える `to_gphys` メソッドが使われている。

サーバとクライアントとの連携の例としてここで `find` メソッドの処理の流れを示す。まず `find` メソッド内部ではリモートの Web サービスに送るための URL を生成し、対象となるホストにアクセスを行う。5 行目の `find` メソッドにおいて発行される URL は以下のとおりである。

`http://host:port/find(path=/data/samples)`

`find` メソッドの記述方法は 4.2 節にて述べる。スクリプトの上では `find` の戻り値 `samples` は GfdnaviArray クラス (GfdnaviData の配列) であるが実際には検索結果を取得するわけではない。クライアントで実際にデータが必要となるまではサーバでの処理結果のメタ情報をやりとりして処理を進めていく遅延処理を取り入れている。4 行目の `find` では検索結果の該当件数と各データのメタ情報 (検索結果データの基本属性 (名前, パス, 説明文など)) または検索に失敗した場合はエラーメッセージが実際には送り返される。5 行目にて再

び find が実行されると、4 行目の find とあわせて以下のような URL が構成される。

```
http://host.port/find(path=/data/samples)/
find(kw.long_name=temperature&sp.overlap(0,0,90,90))
```

このように GfdnaviData および GfdnaviArray オブジェクトの内部では URL が受け渡しされ、サーバで行われる一連の処理を構成する。

8行目でGfdnaviArrayオブジェクトのresultsに対してeach文が実行されるとそのブロックの中はGfdnaviArrayの配列の一つ一つの要素に対する処理となる。ここでは一つ一つの要素をGfdnaviDataオブジェクトtに代入している。この際each文の内部ではfindメソッドで得られた検索結果のメタ情報から各データの基本情報を取得し、そのデータ単体へアクセスするURLに構成しなおす。例えば配列であるresultsの一番最初のオブジェクトの基本情報が以下のとおりだとする。

name : T

path : /data/samples/reanalysis/ncp/T.jan.nc/T

このときGfdnaviDataオブジェクトtは以下のURLを内部に持つ。

```
http://host:port/data/samples/reanalysis/ncp/T.jan.nc/T
```

なお、前述のGPhysは、このようなファイル中の数値データを表すためのRubyのクラスライブラリであり、複数のファイル形式に統一に対応している。

9行目で解析メソッドanalysisを実行し、10行目でto_gphysメソッドを実行する時点で初めてクライアントで実際のデータが必要になる。この時点で発行されるURLは以下のとおりである。

```
http://host.port/data/samples/reanalysis/ncp/T.jan.nc/T/analysis(cut;lon=130).nc
```

このURLでは最後に拡張子「.nc」がついている。この拡張子はNetCDFファイル形式の拡張子であり、to_gphysメソッドはこのデータを受け取りGPhysで扱える形式に直して出力する。これまでのURLでは拡張子が省略されているが、その場合はデフォルトの拡張子yamlがあるとみなし、メタデータをyaml形式でクライアントに戻すようにしている。

以下4.2節にてURLの記述ルールについて、4.3節にてGfdnaviDataおよびGfdnaviArrayクラスについて述べる。

4.2. Gfdnavi Web サービスのリクエスト記述

まず基本的な記述方針について述べる。分岐や結合のない単純なワークフロー（検索→解析→可視化）を

記述する場合は以下のとおりとなる。

```
http://host:port/path/method1(params1)/method2(params2)/....methodn(paramsn).extension?params
```

処理の流れを「/」で区切り、一連の処理の結果を拡張子extensionで指定されたデータ形式で取得する、という要求を記述したものである。データを取得する際のパラメタは拡張子の後に続いて記述する。なおメソッドを一つも使わずに

```
http://host:port/path1
```

表すことも可能である。

上の記述でmethod2はmethod1の出力を入力として受け取り、method2の出力をmethod3に受け渡す。また複数のフローを途中で結合する際には以下のように記述する。

```
http://host:port/[path/method1(params1)/method2(params2), /path/method3(params3)/method4(params4)]/method5(params5).extension?params
```

つまり複数の method(例えば ma(),mb(),mc())による出力をひとつの method の入力とする際には[ma(),mb(),mc()]と書き配列としてあらわす。それ以下の処理は配列に対する処理となる。

サーバで行う処理であるメソッドとして我々は

- find : 検索用メソッド
- analysis : 解析用メソッド
- plot : 可視化用メソッド

を提供する。以下各メソッドのパラメタ記述について述べる。

【findメソッド】

・**入力** : pathで指定されたデータおよびfindの結果、もしくはそれらの配列を入力とする。例えばパス/data/samples下のオブジェクトに対して検索を行う場合は

```
http://host:port/data/sample/find(...)
```

と記述する。現時点にはfindメソッドはanalysisおよびplotの出力を入力とすることができない。

・**出力** : 検索条件に該当するオブジェクトの配列を出力とする。

・**パラメタ** : findメソッドのパラメタは以下のように指定する。

```
find(all or first;検索条件;検索パラメタ)
```

all or first は all と指定した場合検索結果すべてを返し、first と指定した場合は最初の結果のみ返す。

検索条件では検索条件を指定する。条件の表し方は、

キーワード検索 : kw.属性名 = 属性値

フリーキーワード検索 : fw=keyword

空間検索 : sp.overlap=[slon,slat,elon,elat]

データタイプ検索 : datatype=datatype

ディレクトリパス検索 : path=path

時間検索 : `tm=[start,end]`

となる．検索条件が 2 つ以上の場合，例で示されているように検索条件を「&」でつなげる．

検索パラメタは検索条件以外のパラメタを指定する．

現時点では以下のパラメタのみ提供する．

オフセット指定 : `offset = offset`

返す結果の個数を指定 : `limit = limit`

二つとも指定する場合はカンマ「,」でつなげる．

【analysisメソッド】

入力 : `path` で指定されたデータおよび `find` や `analysis` の出力，もしくはそれらの配列を入力とする．

出力 : 解析結果オブジェクトの配列を出力とする．

`analysis` メソッドの入出力は配列であるが，例えば `sub` メソッドが二つの数値データを入力として受け取り一つの出力（計算結果）を返すような演算の場合には，配列の中の前から二つずつを入力にしてメソッドを適用しその結果の配列を出力とする．つまり

`[pathA,pathB,pathC,pathD]/analysis(sub)`

の出力は

`[[pathA,pathB]/analysis(sub),`

`[pathC,pathD]/analysis(sub)]`

と同じ処理を返す．なお，入力が三つのメソッドの場合は，三つずつ処理される．

パラメタ : `analysis` メソッドのパラメタは以下のように指定する．

`analysis(analysis_method;arg0,...,argn)`

`analysis_method` にはメソッド名を指定する．解析メソッドは `Gfdnavi` であらかじめ提供しているもののほか，ユーザ登録も可能である．ユーザが登録した解析メソッドを指定する場合には `analysis_method` の部分には `method,account`

と指定する．ここで `account` とはメソッドを登録したユーザのアカウント名をさす．`arg0,...,argn` はそれぞれの解析メソッドに必要なパラメタを指定する．

【plotメソッド】

入力 : `path` で指定されたデータおよび `find` や `analysis` の出力，もしくはそれらの配列を入力とする．

出力 : 描画結果オブジェクトの配列を出力とする．

パラメタ : `plot` メソッドは `analysis` メソッドとほぼ同様の記述方法である．

`plot(plot_method;arg0,...,argn)`

【拡張子】

拡張子は取得するデータの形式を指定する．指定をしない場合はデフォルトで `html` 形式が指定され，処理

結果のメタデータが出力される．

拡張子部分は

`.extension?params`

と指定する．拡張子後の表示用パラメタは出力をデータとしてクライアントに渡す際のパラメタを指定する．そのパラメタの指定は最後に実行されるメソッドに従う．現時点では `find` メソッドに対して以下の表示用パラメタが存在する．

`kw_facets=1`

検索結果リストに加えてキーワード属性のファセットを出力する

`sp_facets=1`

検索結果リストに加えて空間属性のファセットを出力する

ファセットについては第 5 節にて述べる．

4.3. Gfdnavi Web サービスアクセス用クラス

4.1 節にも述べたように `GfdnaviWeb` サービスへのアクセス用に `GfdnaviData`, `GfdnaviArray` クラスを提供している．

- `GfdnaviData.parse("http://host:port/path")`

指定されたホストとパスに対応するオブジェクトを取得する．

- `GfdnaviData#find(all-or-first, conditions, query-parameters)`

`GfdnaviArray#find(all-or-first, conditions, query-parameters)`

ここで `#` はインスタンスメソッドであることを示す．

`conditions` で指定された検索条件に該当するオブジェクトを `all-or-first, query-parameters` の指定に従って検索する．各パラメタの記述は Web サービスの `find` メソッドと同様である．

- `GfdnaviData#analysis(analysis_method,[account];arg0,...,argn)`

`GfdnaviArray#analysis(analysis_method,[account];arg0,...,argn)`

- `GfdnaviData#plot(plot_method,[account];arg0,...,argn)`

`GfdnaviArray#plot(plot_method,[account];arg0,...,argn)`

ここでの `[]` は省略可能であることを示す．

解析, 可視化メソッドも検索と同様に各パラメタの記述は Web サービスの `analysis, plot` メソッドに従う．

- `GfdnaviData#to_extension(params)`

`GfdnaviData` を `extension` 形式で出力する．`extension` には `yaml`, `xml`, `gphys`, `nc`, `html` 等があり，どれを指定できるかは `GfdnaviData` がどのメソッドから出力されたかに依存する．`params` は Web サービスの表示用パラメタに従う．

5. GfdnaviWeb サービスの一応用：横断検索

Gfdnavi の横断検索が実現すれば、数値データ・解析データ・解析手法・知見情報などの検索をサーバ間で行えるようになり、利用目的に応じたデータの検索や実践的な研究ノウハウの検索が可能となる。本節では GfdnaviWeb サービスを利用して少数のサイトにまたがった Gfdnavi 横断検索を実現する。

Gfdnavi では、検索インタフェースに Faceted Navigation という検索手法を採用している。したがって、横断検索でもこの検索インタフェースを利用し、ユーザが通常の検索を行っているような感覚でサーバ間の検索が行える横断検索を提供できるように Web サービスの拡張を行った。検索インタフェース、Web サービスの拡張については、5.2 節以降で述べる。

5.1. 横断検索の検索インタフェース

まず Gfdnavi にて提供している検索インタフェース Gfdexplorer について述べる。スクリーンショットを図 7 に示す。

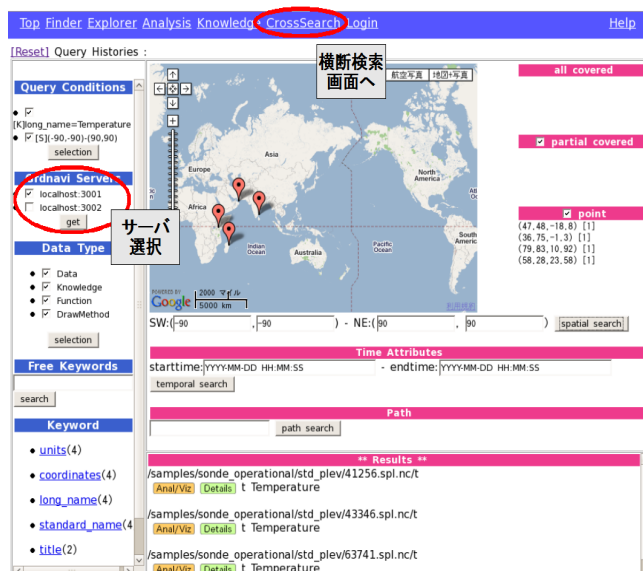


図 7：Gfdexplorer のスクリーンショット

検索条件としてはデータタイプ検索、フリーキーワード検索、キーワード検索、空間条件、時間条件を指定できる。データタイプ検索では、検索したいデータの種類を選択することができる。データの種類には、データ・知見情報・関数・解析手法などがある。フリーキーワード検索は、ユーザが検索条件となるキーワードを入力して検索を行う。その下のキーワード検索では、その時点での検索結果オブジェクト集合が持つキーワード属性(属性名と属性値)をリストアップし、各属性値を持つオブジェクト数を表示している。ユーザはリストから興味のある属性を選択するだけで絞込検索を行うことができる。このような検索結果の情報

を提示しながら対話的に絞込みを行う検索モデルを「ファセットナビゲーション」と呼ぶ。ファセットナビゲーションでは各属性をファセットとよび、ファセットのとりうる値のリストつまり属性値のリストと該当件数を提示する。検索の際に用いた検索条件は、Gfdexplorer の左上の Query Conditions 欄に表示される。キーワード検索なら[K]，空間検索なら[S]，フリーキーワード検索なら[F]，データタイプ検索なら[N]，時間検索なら[T]，ディレクトリパス検索なら[P]が頭に表示される。各条件の左にチェックボックスが設けられ、ボックスのチェックをつけたりはずしたりすることで、条件を選択することが可能となる。

5.2. Web サービスによる横断検索の処理の流れ

5.1 節に述べた検索インタフェースで複数の Gfdnavi サーバへ検索を行い、その結果をまとめて提示するための処理の流れを図 8 に示す。

まず Gfdnavi 公開サーバリスト情報を持つ中央サーバを立ち上げる。検索を行う際には公開サーバの情報を中央サーバに要求して取得する。そして、サーバ情報を用いて公開サーバを指定し GfdnaviData クラスの find メソッドを実行して各サーバで検索を行う。また、中央サーバに登録されている公開サーバ以外にも個人的に URL を知っている Gfdnavi サーバがあれば、検索対象のサーバとして追加することができるようにする。

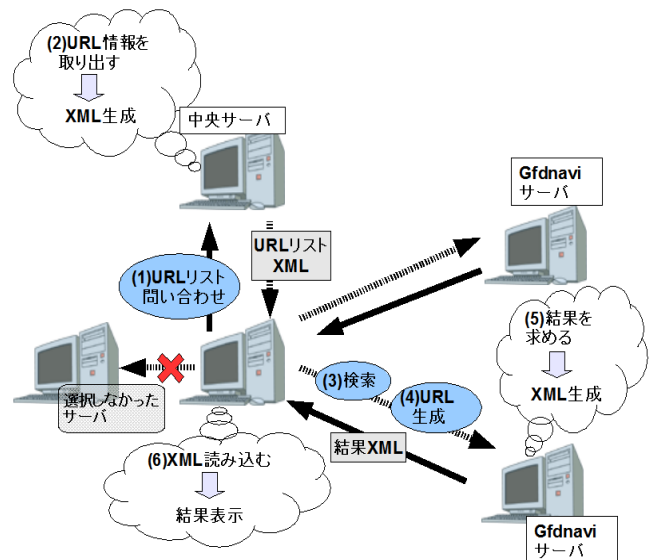


図 8：想定される横断検索の状況
(中央サーバ以外はすべて Gfdnavi サーバで構成)

それぞれの処理は以下のように分けられる。

1. サーバ URL リストの受信 (図 8 の(1), (2))
2. 各 Gfdnavi サーバへの検索リクエストを生成 (図 8 の(3), (4))
3. 他サーバが検索結果を生成し, XML 形式に変換 (図 8 の(5))

4. XML 形式の検索結果を取得し,その結果を読み込んで表示 (図 8 の(6))

5.3. 実行例

以下に属性名 long_name,属性値 Temperature かつ (-90,-90)-(-90,90)の空間検索を 3 つのサーバ

hostA:portA, hostB:portB, hostC:portC に発行し結果を得るときのスクリプトの一部を示す.

```
rootA=GfdnaviNavi.parse("http://hostA:portA")
rootA.find("all",kw.long_name=Temperature&sp.overlap
[-90,-90,90,90"]) .to_xml("kw_facets=1&sp_facets=1")
```

```
rootB=GfdnaviNavi.parse("http://hostB:portB")
rootB.find("all",kw.long_name=Temperature&sp.overlap
[-90,-90,90,90"]) .to_xml("kw_facets=1&sp_facets=1")
```

```
rootC=GfdnaviNavi.parse("http://hostC:portC")
rootC.find("all",kw.long_name=Temperature&sp.overlap
[-90,-90,90,90"]).to_xml("kw_facets=1&sp_facets=1")
```

これを実行することにより hostA,B,C から検索結果とファセットリストを xml 形式で収集し,取得した結果 XML から DOM ツリーを生成して結果をそれぞれ読み込む.そして,それらを結果セットとして Gfdexplorer に表示する. XML 形式で書かれた結果とそれを取得して読み込んだ画面は図 9 のようになる.



図 9: XML 形式の属性値・該当数 (左) と XML を読み込んだ画面 (右)

6. まとめと今後の課題

本稿では, Gfdnavi においてデータの検索・解析・可視化を行う REST-style の Web サービスの提供につ

いて述べた. また, その Web サービスを使い複数のサーバに対する検索の実装を行った. 今後は, 横断検索プログラムの改良や, より低コストで提供できるように実装していく予定である. また, サーバごとに結果が分かるような工夫を行うなど, ユーザにとって使いやすいツールになるように実装を行っていく.

謝辞

本研究は, 文部科学省科研費特定領域「情報爆発時代に向けた新しい IT 基盤技術の研究」の課題(課題番号 19024039)により行われた. 本研究遂行にあたって様々な協力を頂いた地球流体電脳倶楽部の各氏に感謝する.

参考文献

- [1] 地球流体電脳倶楽部
<http://www.gfd-dennou.org/>
- [2] 堀之内武,西澤誠也, 渡辺知恵美, 森川靖大, 神代剛, 石渡正樹, 林祥介, 塩谷雅人, “地球流体データベース・解析・可視化のための新しいサーバ兼デスクトップツール Gfdnavi の開発”, DEWS 2007.
- [3] 重川美咲子, 西澤誠也, 堀之内武, 渡辺知恵美, “Gfdnavi: 地球流体物理学者のためのデータアーカイブサーバ構築支援ツールデータ属性の探索的検索を利用する検索”, DEWS 2008.
- [4] 地球流体電脳倶楽部
<http://www.gfd-dennou.org/library/davis/products/index.htm>
- [5] GSMaP
http://www.radar.aero.osakafu-u.ac.jp/~gsmap/index_japanese.html
- [6] 伴林晃紀, 堀之内武, 津田敏隆, 渡辺知恵美, 西澤誠也, “地球流体データ解析・可視化ツール Gfdnavi における知見情報のデータベース化”, DEWS 2008.
- [7] 齋藤真衣, 堀之内武, 渡辺知恵美, “対話的な絞り込み操作を考慮した P2P による地球流体データアーカイブサーバの横断検索”, DEWS 2008.