Gfdnavi, Web-Based Data and Knowledge Server Software for Geophysical Fluid Sciences, Part I: Rationales, Stand-Alone Features, and Supporting Knowledge Documentation Linked to Data

Takeshi Horinouchi[1], Seiya Nishizawa[2], Chiemi Watanabe[3], Akinori Tomobayashi[4], Shigenori Otsuka[5], Tsuyoshi Koshiro[6], Yoshi-Yuki Hayashi[2], and GFD Dennou Club[7]

[1] Faculty of Environmental Earth Science, Hokkaido University, N10W5 Sapporo, Hokkaido 060-0810, Japan
[2] Department of Earth and Planetary Sciences, Kobe University
[3] Department of Information Sciences, Ochanomizu University
[4] Shouganji
[5] Department of Geophysics, Kyoto University
[6] Climate Research Department, Meteorological Research Institute
[7] www.gfd-dennou.org

Abstract. In recent years, many data centers and research groups provide data on geophysical fluids such as the atmosphere and oceans through the Internet along with on-line visualization. However, their services are not available once data files are downloaded. This paper presents open-source software named Gfdnavi developed to reduce the limitation and to support data handling beyond initial "quick-looks". Gfdnavi extracts metadata from scientific data and stores them in a database. They can be accessed with web browsers for search, analysis, and visualization. It supports a wide range of usage such as public data services, group data management, and desktop use. As its unique feature, Gfdnavi supports writing and archiving documents based on knowledge obtained through data analysis. The documents are linked with the original data and analysis/visualization procedures. It has a wide variety of applications such as interdisciplinary- and collaborative-study support, a realization of falsifiability, and educational use.

Key words: Data server, Geophysical fluid sciences, Visualization, Web application, Knowledge archive

1 Introduction

Contemporary scientists of "geophysical fluids", such as the atmosphere and the ocean, are facing a rapid increase of data. Observational data from satellites, ground-based remote-sensing instruments, and *in situ* measurements have been increasing year by year in both amount and kinds. Also, numerical simulations, such as climate prediction, produce vast amount of data. Therefore, to enhance information technology (IT) infrastructure for data access and handling would be helpful for the sciences.

There is also increasing demand for interdisciplinary studies to solve environmental problems. In such studies scientists have often to use data from fields of studies that are not familiar to them. Thus, IT infrastructure would also be needed to assist them.

Recently, many research organizations provide geophysical fluid data through the Internet. Some provides server-side visualization as well. Although many data servers are still custom made, it is becoming common to use data-server construction tools such as Live Access Server (LAS)[1]. LAS is the most advanced tool used for geophysical fluid sciences. A LAS server enables its users to visualize data using web browsers. It is highly configurable so that its administrator can design a façade suitable to its contents.

There are, however, still problems with currently used data servers for geophysical fluid sciences as follows:

1. Visualization capability is generally limited, so only "quick-looks" on the initial stage of researches are possible.
2. The features of the servers are not available once data files are downloaded. Therefore, the data need to be analyzed and visualized independently. In such a case, even to opening the files could be time-consuming, since a number of binary data formats are used in these sciences.
3. Most of the browser-based advanced data servers support only georeferencing data, despite the fact that non-georeferencing data are also important and frequently required for conducting researches.
4. The search capability of these servers is often limited. Furthermore, it is difficult to search for data across network, except for their documents that are available via Internet search engines.
5. Interdisciplinary and/or collaborative studies would require communication among scientists for the purpose of exchanging know-hows, results, and so on. However, to the best of authors' knowledge, none of the existing data servers for geophysical fluid sciences support such communication.

To solve all these problems, we have developed software named *Gfdnavi* [2], which stands for Geophysical fluid data navigator. Our approach is to create software that seamlessly support a variety of use cases from the management and analysis of local (desktop) data by individual scientists to public data services, where one can not only download but also interactively analyze and visualize data.

An alternative approach to help remote data access is to provide a good library to support client-side data analysis and visualization. In geophysical fluid sciences, it is OPeNDAP [3] that is widely used for such purposes. OPeNDAP provides remote access to numerical data situated in directory trees. The OPeNDAP server is a CGI program and is frequently installed on web servers of public data centers to allow data access through application programming interfaces (APIs). OPeNDAP has an abstract data model to hide file formats and supports subsetting. The two approaches, client-side analyses with OPeNDAP and server-side analyses with LAS or Gfdnavi, can complement each other.

This paper describes the design and features of Gfdnavi as a stand-alone web application. The companion paper [4] describes its web services and synergetic use across multiple Gfdnavi servers. The rest of the paper is organized as follows. Design principles and decisions made are explained in Section 2. Section 3 introduces Gfdnavi as a stand-alone data server. In Section 4, we argue the usefulness to support collection of knowledge that users obtained through data analysis, and we introduce its implementation. Conclusions are drawn in Section 5.

2 Design Rationales

Gfdnavi is designed and implemented to solve the problems stated in the previous section. In this section we introduce the principles and decisions made when designing it.

Since it is impossible to predefine all possible scientific data-analysis and visualization functionalities, in order to solve the problem 1 it is necessary to provide some kind of programmability for clients. We chose to provide programmability in multiple ways as follows:

1. To support web services.
2. To allow its user to download a subset of data and a script to reproduce what is conducted on the server. Then, one can further refine the analysis and/or visualization.
3. To allow its registered users to upload scripts to conduct data analysis and visualization on the server.

Here, we chose the Ruby programming language as the primary language to provide programmability, so the second and the third of the programmability are implemented for this language. Also, we provide a Ruby library to access the web service as introduced by [4].

The language choice is justified as follows. We earlier developed a Ruby class library GPhys [5], which represents multi-dimensional numerical data and is used by a growing number of scientists to conduct geophysical fluid data analysis and visualization. It supports a variety of data formats and solves the problem 3, so it is quite suitable for our purposes. By extensively supporting programmability, we can bridge the gap between public data services and desktop data handling (the problem 2) in terms of programming.

Our design principle in providing programmability is to unify application programming interfaces (APIs) irrespective of data location (whether they are on run-time memory or in locally accessible external files or over network) and access method (whether through Gfdnavi or other network data services such as

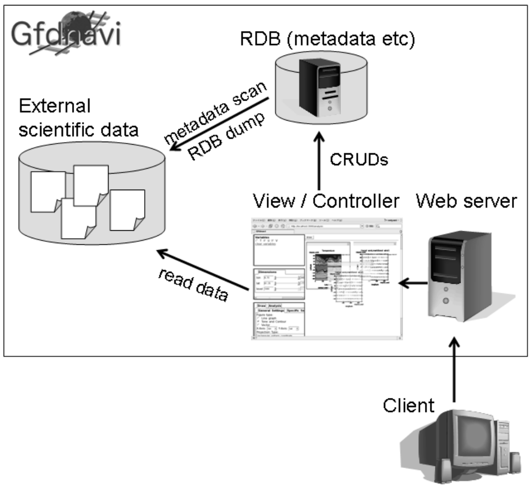


Fig. 1. Overview of the system components of Gfdnavi

OPeNDAP or local IO calls). Ruby is an object-oriented language suitable for such unification.

The problem 2 still remains in terms of browser access. That is, one can access public data servers with a web browser, but he/she cannot use the same browser interface to access local data. The problem can be solved if he/she installs the same data server software locally. To make it practical, the server software should be easy to install, manage, and run for scientists, who are often unfamiliar with web server installation and database management systems. Gfdnavi is designed as such.

The problem 4 should also be treated. Search among a single Gfdnavi server is introduced in what follows, and search across multiple Gfdnavi servers is introduced by the companion paper [4].

Last but not least, to tackle the problem 5 is important. Other data servers only provide access to data and do not have features to collect knowledge obtained through data analysis. We propose to support document creation on data servers, in which data analysis processes are automatically recorded and saved with the documents created. As shown in Section 4, a variety of applications can be made possible by inter-linking scientific data, analysis and visualization procedures, and the documents. This is a unique feature of Gfdnavi.

3 Overview of Gfdnavi as a data server

Gfdnavi is a set of software to build and run a data server. Figure 1 shows an overview of the system components of Gfdnavi. A user of a Gfdnavi server accesses it with a web browser or through web service APIs. The server serves data in local file systems or on remote servers for analysis and visualization. It

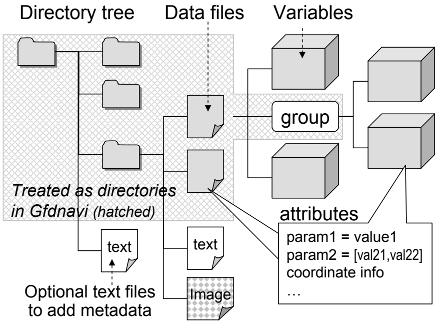


Fig. 2. A schematic illustration of the directory tree (on local storage or remote OPeNDAP servers) to be served by Gfdnavi.

also serves metadata, stored in a relational database (RDB), for browsing and search.

Gfdnavi was developed with the Ruby on Rails application development framework [6][7], which helps develop web applications that utilize RDBs. It supports most of major RDB managements systems (RDBMSs). Owing to the use of the framework, Gfdnavi is equipped with a custom web server program, which runs on arbitrary communication port with a single command. Therefore, one can run a Gfdnavi server easily on a PC whenever needed, which makes it easier to use Gfdnavi personally on a PC than other data-server tools such as LAS. Gfdnavi can also be operated with commonly used web servers such as Apache, which is suitable to long-term public data services. Since most scientists are not familiar with RDBMSs, the installer of Gfdnavi suggests to use SQLite, a non-daemon type RDBMS, if a desktop configuration is selected, which makes it easy to use Gfdnavi furthermore.

Basic features of Gfdnavi are available for anonymous users. However, it has a "login" system in order to allow access limitation and to support features that are not necessarily safe, such as uploading source codes, or computationally demanding. The basic account system is local to each Gfdnavi server, supposing that the administrators give accounts only to reliable persons. In addition, Gfdnavi supports OpenID[8] for authentication that is available across servers. For security, OpenID accounts are restricted, so, for example, uploading source codes is prohibited.

3.1 Scientific data and metadata database

To start up or maintain a data server, Gfdnavi collects metadata from data files under specified directories in local file systems or on remote OPeNDAP servers (Fig. 2). Most scientific data formats used in geophysical fluid sciences such as NetCDF [8] have metadata embedded. In addition, Gfdnavi supports texts files

[8] http://openid.net/

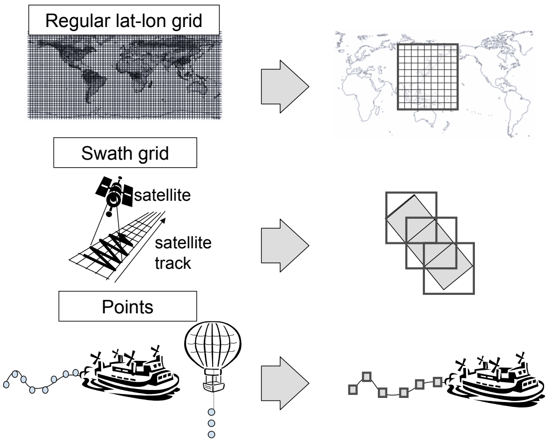


Fig. 3. Treatment of a variety of spatial sampling by bounding boxes. Irregular sampling in terms of longitude and latitude are bounded by multiple boxes, as shown by thick gray lines in the rhs panels.

to supplement metadata. The metadata collected are stored in a RDB along with the structure of directory trees.

The metadata of numerical data consists of keyword attributes and space-and-time attributes. A keyword attribute is a combination of a name and a value (or values). A space-and-time attribute is a three-dimensional bounding box in terms of longitude, latitude, and time. As shown in Fig. 3, this simple strategy covers a variety of sampling by allowing to have multiple bounding boxes.

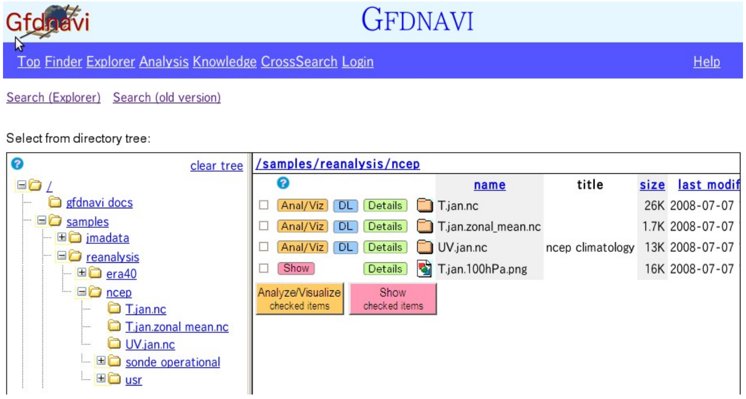
Note that Gfdnavi can handle data without georeferencing and/or time dimensions. Such data are simply excluded from searches based on space and/or time, but they can be found by keyword search or in the directory tree. As for numerical analysis and visualization, any kinds of coordinates are accepted. For example, spectral data that are functions of frequencies are accepted.

Scientists normally organize data directories hierarchically, so metadata of a directory are likely to be applicable to its children. Therefore, in order to make automatically generated metadata practical, we treat metadata to be inherited downward the tree. This treatment is useful to supplement data files with poor metadata. Because of this downward inheritance, metadata in Gfdnavi can be supplemented efficiently by placing text files at right levels in the directory hierarchy.

The downward inheritance may not necessarily be adequate. However, a good ranking and user interface (UI) could allow screening of false match. It is, therefore, rather important not to exclude data wanted.

3.2 Browser user interface

Finding data on a Gfdnavi server using web browser is based on two UIs: a directory tree viewer and a search window. Figure 4 shows a screen-shot of the



				name	title	size	last modif
□	Anal/Viz	DL	Details	T.jan.nc		26K	2008-07-07
□	Anal/Viz	DL	Details	T.jan.zonal_mean.nc		1.7K	2008-07-07
□	Anal/Viz	DL	Details	UV.jan.nc	ncep climatology	13K	2008-07-07
□	Show		Details	T.jan.100hPa.png		16K	2008-07-07

Fig. 4. A screen-shot of the directory-tree viewer. By clicking the the "Anal/Viz" buttons, variables under the files are put in a "cart", and the browser is directed to the visualization/analysis page.

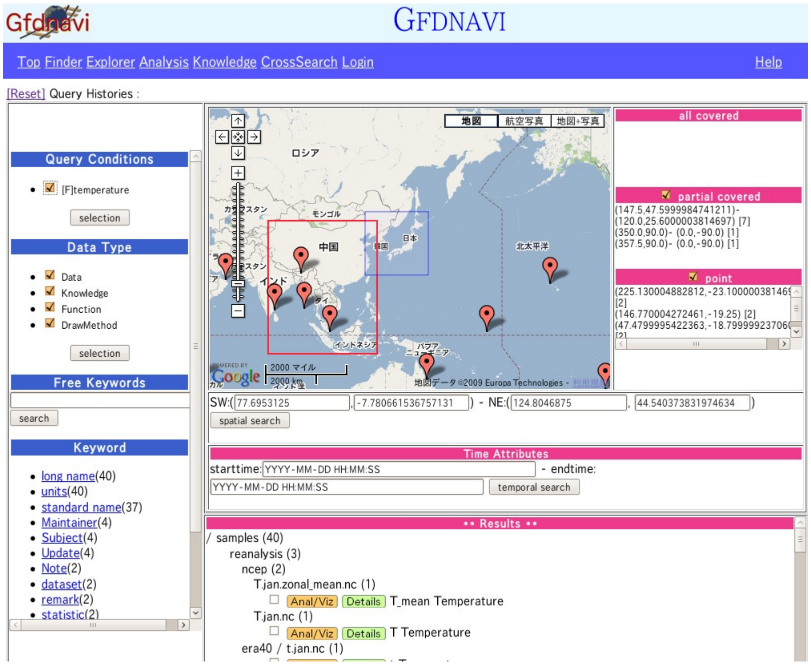
directory tree viewer. With this page, one can browse and select data to analyze and/or visualize. The tree viewer uses asynchronous communication and caching, so a huge data tree with numerous files is handled lightly.

Figure 5 shows a screen-shot of the search window. It provides a variety of searches: free text search, faceted navigation of keyword attribute search, search by location and time. Here, multiple queries can be combined. Also, data types such as numerical data and knowledge documents can be specified.

Numerical data selected in either of the interfaces mentioned above can be analyzed and visualized in a single UI. Figure 6 shows a screen-shot of the UI. Windows on the left-hand-side are the UI to specify visualization ("Draw" tab in the lower left panel) or numerical/statistical analysis ("Analysis" tab). The upper-left window is a "cart" of selected numerical variables. The middle-left panel shows dimensions in the variable and is used for subsetting. The lower-left panel is used to specify various parameters for graphics, including animation.

After visualization, as shown Fig. 7, one can

- download the Ruby script and the minimum subset of data to reproduce the visualization,
- save the image on the server (login required),
- get the URL, with which a window to redo the visualization can be obtained, and
- create a document with figures that consist of visualization results (see Section 4).



Query Histories :

- ☑ [F]temperature

- ☑ Data
- ☑ Knowledge
- ☑ Function
- ☑ DrawMethod

- long name(40)
- units(40)
- standard name(37)
- Maintainer(4)
- Subject(4)
- Update(4)
- Note(2)
- dataset(2)
- remark(2)
- statistic(2)

☑ partial covered
(147.5,47.5999984741211)-(120.0,25.6000003814697) [7]
(350.0,90.0)- (0.0,-90.0) [1]
(357.5,90.0)- (0.0,-90.0) [1]

☑ point
(225.130004882812,-23.10000038146
[2]
(146.770004272461,-19.25) [2]
(47.4799995422363,-18.79999923706
[2]

SW:(77.6953125, -7.780661536757131) - NE:(124.8046875, 44.540373831974634)

starttime: YYYY-MM-DD HH:MM:SS - endtime: YYYY-MM-DD HH:MM:SS

```
/ samples (40)
    reanalysis (3)
      ncep (2)
        T.jan.zonal_mean.nc (1)
          □ Anal/Viz Details T_mean Temperature
        T.jan.nc (1)
          □ Anal/Viz Details T Temperature
      era40 / t.jan.nc (1)
```

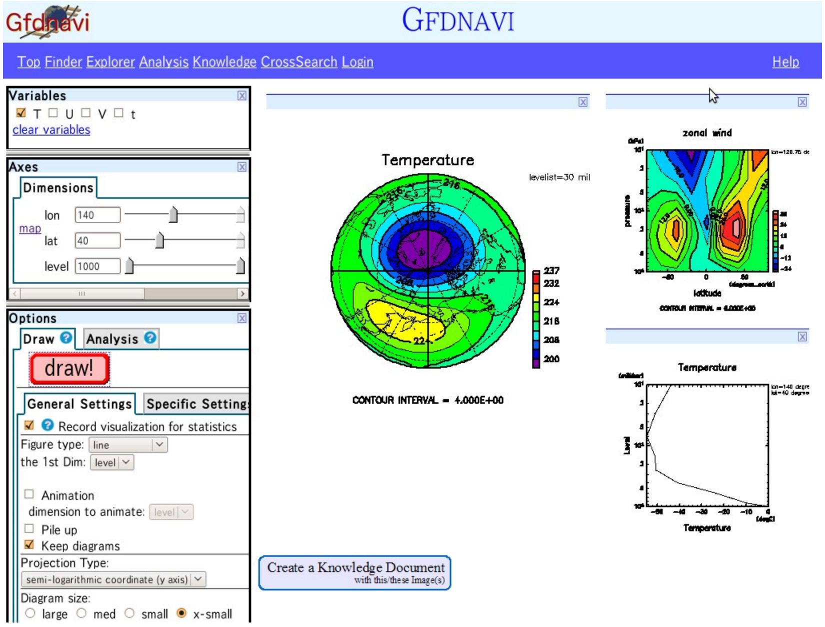
Fig. 5. A screen-shot of the search window.

4 Supporting a knowledge archiving system in a data server

The cycle of a scientific data analysis of geophysical fluids would typically start with visualizing and analyzing numerical data, saving graphics obtained, which is often along with memos, to end by writing a report or paper to publish. It is meaningful to support the entire cycle with one database application, since cross referencing would be available for all the entities involved in the cycle.

Our scope is to support personal research memos, technical documentation, exchange of knowledge and know-hows for interdisciplinary and collaborative studies, outreach or public relations of data centers (such as to create "What's new" pages), and educational materials. We do not aim to support peer-reviewed electronic journal publication systems within Gfdnavi, but in future supplementary materials of research papers could include URLs on Gfdnavi.

A knowledge document in Gfdnavi consists of a title, a summary, a text body, and figures, along with attributes such as category. A figure has not only a caption and an image file but also the script used to create the image. Therefore, if one finds a knowledge document on a Gfdnavi server. He or she can follow links in its figures to reproduce them and further to visualize data with different conditions or even to make additional analysis. This feature is useful to support *falsifiability*, which is important for sciences.



Gfdnavi
GFDNAVI
Top Finder Explorer Analysis Knowledge CrossSearch Login
Help
Temperature
levelist=30 mil
CONTOUR INTERVAL = 4.000E+00
zonal wind
latitude
CONTOUR INTERVAL = 4.000E+00
Temperature
Level
Temperature
Create a Knowledge Document
with this/these Image(s)

Variables

☑ T □ U □ V □ t

clear variables

Axes

Dimensions

lon 140

map

lat 40

level 1000

Options

Draw | Analysis

draw!

General Settings | Specific Settings

☑ Record visualization for statistics

Figure type: line

the 1st Dim: level

□ Animation

dimension to animate: level

□ Pile up

☑ Keep diagrams

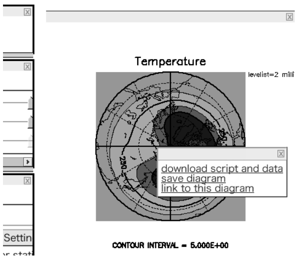
Projection Type:

semi-logarithmic coordinate (y axis)

Diagram size:

○ large ○ med ○ small ● x-small

Fig. 6. A screen-shot of the visualization/analysis UI.



download script and data

save diagram

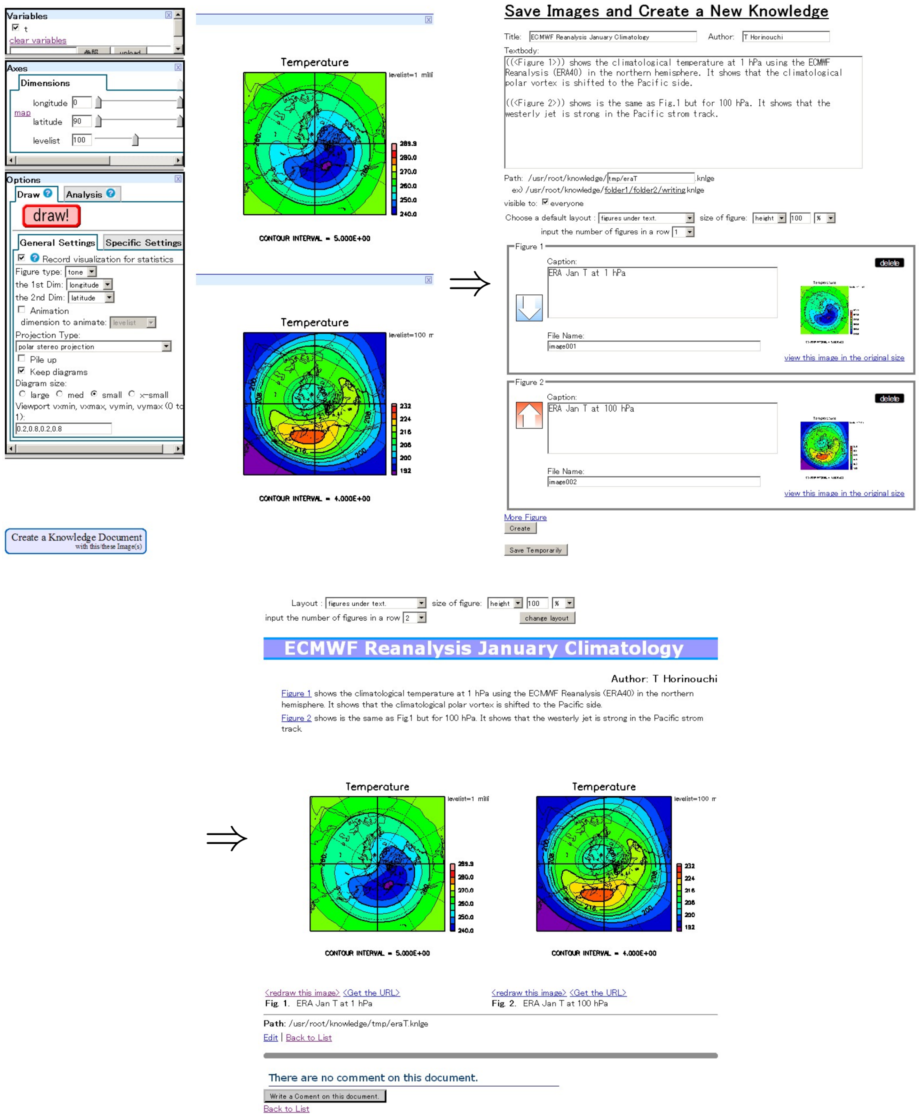
link to this diagram

Fig. 7. A pop-up window to navigate user for further operations after visualization.

This feature is also useful for collaborative and/or interdisciplinary studies. For example, one can document an example of data analysis, with which actual operations may be reproduced. Then its readers can directly follow or extend it.

In order to further support interactions, knowledge documents can take "comments", which are also knowledge documents. With this feature, collaborators can make threads of interactions, where each of the comments can have dynamic links to data.

The knowledge documentation system of Gfdnavi brings a new depth to outreach and education. Many data centers have "what's new" pages or educational materials, in which scientific outcomes are documented for broad readers. By using the knowledge documentation system of Gfdnavi, such a documents can have



longitude 0
latitude 90
levelist 100

☑ Record visualization for statistics
Figure type: tone
the 1st Dim: longitude
the 2nd Dim: latitude
☐ Animation
dimension to animate: levelist
Projection Type:
polar stereo projection
☐ Pile up
☑ Keep diagrams
Diagram size:
○ large ○ med ● small ○ x-small
Viewport vxmin, vxmax, vymin, vymax (0 to 1):
0.2,0.8,0.2,0.8

Save Images and Create a New Knowledge

Title: ECMWF Reanalysis January Climatology
Author: T Horinouchi
Textbody:
((<Figure 1>)) shows the climatological temperature at 1 hPa using the ECMWF Reanalysis (ERA40) in the northern hemisphere. It shows that the climatological polar vortex is shifted to the Pacific side.

((<Figure 2>)) shows is the same as Fig.1 but for 100 hPa. It shows that the westerly jet is strong in the Pacific strom track.

Path: /usr/root/knowledge/tmp/eraT.knlge
ex) /usr/root/knowledge/folder1/folder2/writing.knlge
visible to: ☑ everyone
Choose a default layout : figures under text. size of figure: height 100 %
input the number of figures in a row 1

Figure 1
Caption: ERA Jan T at 1 hPa
File Name: image001

Figure 2
Caption: ERA Jan T at 100 hPa
File Name: image002

Layout : figures under text. size of figure: height 100 %
input the number of figures in a row 2

ECMWF Reanalysis January Climatology

Author: T Horinouchi

Figure 1 shows the climatological temperature at 1 hPa using the ECMWF Reanalysis (ERA40) in the northern hemisphere. It shows that the climatological polar vortex is shifted to the Pacific side.

Figure 2 shows is the same as Fig1 but for 100 hPa. It shows that the westerly jet is strong in the Pacific strom track.

Temperature
levelist=1 milli
283.3
280.0
270.0
260.0
250.0
240.0
CONTOUR INTERVAL = 5.000E+00
Temperature
levelist=100 m
232
224
216
208
200
192
CONTOUR INTERVAL = 4.000E+00
<redraw this image> <Get the URL>
<redraw this image> <Get the URL>

Fig. 1. ERA Jan T at 1 hPa
Fig. 2. ERA Jan T at 100 hPa

Path: /usr/root/knowledge/tmp/eraT.knlge

There are no comment on this document.

Fig. 8. Demonstration of knowledge document creation from visualization results. By clicking the rounded rectangle below at the bottom of the visualization window (upper-left panel), the user is lead to a document input form (lower-left panel), in which the images on the visualization window are set to figures by default. The lower panel shows the document created.

figures that are linked to the data used and are dynamically reproducible. The readers can follow links from the figures to further explore scientific data by themselves.

The knowledge documents annotate numerical data. If scientists work with Gfdnavi and stores what they find as knowledge documents, the numerical data used are annotated with text and visual information. Therefore, it can be regarded as a system to collect high-level metadata from scientists. For example, one may leave a document on a specific typhoon found in year 2009 using an image made from a global rainfall dataset based on satellite-borne microwave radiometers. Since the image is saved with visualization parameters, it automatically has space-time information on the typhoon. In other words, saving the document annotates the rainfall data with text, image, and space-time information on the typhoon. The annotation enriches the metadata of the dataset significantly, since the original metadata of microwave rainfall data may not even have the word “typhoon”. In terms of data search, one may find the microwave radiometer data useful to visualize typhoons because of this document. This is useful especially for those whose specialty is not the atmospheric sciences.

Figure 8 demonstrates how a knowledge document can be created from visualization results. By clicking a button on the visualization window (upper-left panel), the user is lead to a document input form including the visualization results as default figures (lower-left panel). By using the form, visualization procedures are recorded automatically. The procedures to reproduce the visualization from the original data are recorded automatically, and the document created have links to them (bottom panel).

The documents made with Gfdnavi can be browsed and searched in many ways. They are covered by the search UI introduced in Section 3. In addition, Gfdnavi has a window specialized for document browsing and search.

5 Conclusions

In this paper, we presented the problems that the web-browser-based data servers for geophysical fluid sciences have, and we introduced our software Gfdnavi to solve them. The gap between network data services and desktop data handling can be solved by extensively supporting programmability and by making the server software easy to install, manage, and run. A unique feature of Gfdnavi is to support writing and archival of documents based on knowledge that its users obtained through data analysis. It has a wide variety of applications such as accumulation of knowledge, interdisciplinary- and collaborative-study support, a realization of falsifiability, and creation of dynamic documents for outreach and education.

Acknowledgments. This study was supported by the MEXT grant-in-aid “Cyber infrastructure for the information-explosion era” A01-14 (19024039, 21013002). We thank Masato Shiotani, Masaki Ishiwatari, Masatsugu Odaka, and Tomohiro Taniguchi for supporting and promoting this work. We thank Takuji Kubota, Kazuki Joe, Yumi Yanagitaira, Asami Sato, Mai Saito, Eriko Touma, Toshitaka Tsuda, Yasuhiro Morikawa, Youhei Sasaki, Eriko Nishimoto, and Mariko Horikawa for their contributions and comments.

References

1. Hankin, S., J. Callahan, J. Sirott: The Live Access Server and DODS: Web visualization and data fusion for distributed holdings. http://ferret.wrc.noaa.gov/Ferret/LAS/LASoverview.html (2001)
2. Horinouchi, T., S. Nishizawa, C. Watanabe, and collaborators: Gfdnavi homepage. http://www.gfd-dennou.org/arch/davis/gfdnavi/
3. Cornillon, P., J. Gallagher, and T. Sgouros: "OPeNDAP: Accessing data in a distributed, heterogeneous environment". Data Science Journal **2** (2003) 164–174
4. Nishizawa, S., T. Horinouchi, C. Watanabe, Y. Isamoto, A. Tomobayashi, S. Otsuka, GFD Dennou Club: Gfdnavi, Web-based Data and Knowledge Server Software for Geophysical Fluid Sciences, Part II: Web services. *submitted to Data Intensive eScience Workshop*, 2010.
5. Horinouchi T., R. Mizuta, S. Nishizawa, D. Tsukahara, and S. Takehiro: GPhys – a multi-purpose class to handle gridded physical quantities. http://ruby.gfd-dennou.org/products/gphys/ (2003)
6. Hansson, D. H. et al.: Ruby on Rails. http://www.rubyonrails.org/
7. Thomas, D., and D.H. Hansson: Agile Web Development with Rails, The Pragmatic Programmers LLC, USA (2005)
8. Rew, R., and G. Davis; NetCDF – An interface for scientific-data access. IEEE Computer Graphics and Applications, **10**(4), (1990) 76–82