

Gfdnavi: A tool to archive, share, distribute, analyze, and visualize geophysical fluid data and knowledge



**Takeshi Horinouchi** (Kyoto Univ → Hokkaido Univ(soon)),

**Seiya Nishizawa** (Kyoto Univ → Kobe Univ),

Chiemi Watanabe (Ochanomizu Univ),

T. Koshiro, A. Tomobayashi, S. Otsuka,  
Y. Morikawa, Y.-Y. Hayashi, M. Shiotani, and

***GFD Dennou Club (Davis project)***

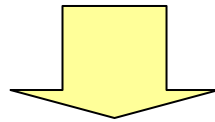
# What is Gfdnavi

- = Geophysical fluid data navigator
- A suite of software to construct Web-based database of geophysical fluid data
- Functionality:
  - Search
  - Data analysis and visualization
  - Documentation of analysis results
- Available:  
<http://www.gfd-dennou.org/arch/davis/gfdnavi/>

Background

# Problems of Web-based database and analysis tools

- Limited analysis capability
  - We often end up with downloading data
- Not very suitable to desktop use
  - Service are not available to local data



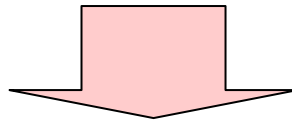
We would rather like to extend desktop tools (such as IDV) to cover persistent data services

# More on the analysis capability

- Impossible to predefine sufficient functionality (since we are scientists)
  - Programmability is the key
- Programmability in two ways:
  - Programmable on web-browser
  - Web-service API (program locally)
- Both are desirable

# Visualization is not the goal

- To others (scientists / society): reports
- While working: memos / internal documents
- To collaborators: reports / know-how / discussion



Outputs are documents  
(not just pieces of images)

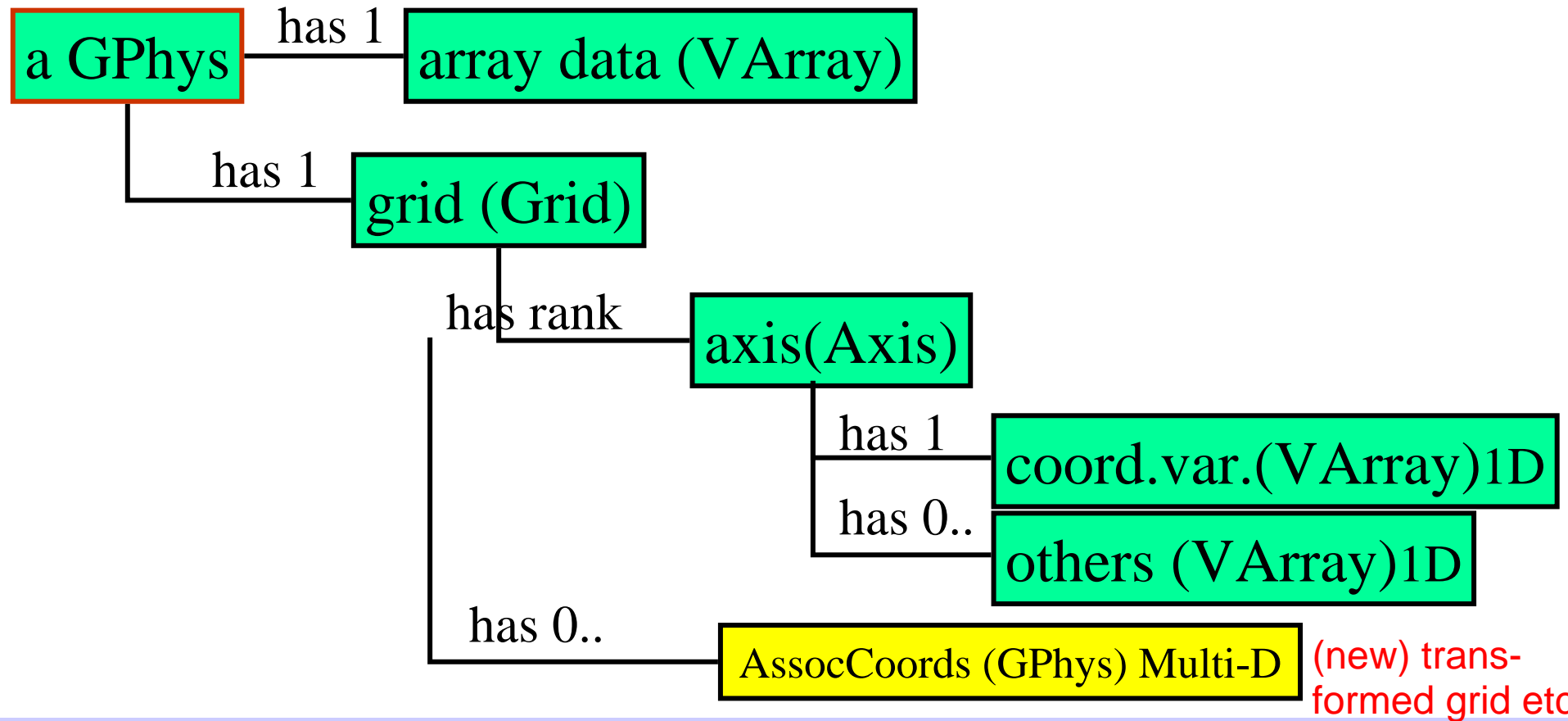
# Foundation of Gfdnavi

# Two fundamental libraries used to build Gfdnavi (*open-source*)

- **GPhys** – a Ruby library to analyze and visualize geophysical fluid data (*by Horinouchi etc since 2003*)
  - For consolidated access to data in files (NetCDF, GRIB, GrADS, NuSDAS, HDF5-EOS) or on runtime memory – **A community infrastructure for data analysis** [<http://ruby.gfd-dennou.org/>] (*since 1999*)
- **Ruby on Rails** – Development framework for Web application (*since 2005*)
  - Made it **drastically easy** to develop Web applications with **RDB**
  - Written in/for Ruby → **We can use GPhys directly**



# GPhys (Gridded Physical quantity)



**VArray (Virtual Array)** – with attributes (incl. units)

Abstracts Data Storage

(Entity can be in file(s) or multi-D Array on memory; can also be a mapped subset of another VArray or aggregation of VArrays)

```
u = GPhys::IO.open("u.nc","U")      ← in NetCDF [m/s]
v = GPhys::IO.open("v.ct1","V")     ← in GrADS [m/s]
uv = u * v                          ← result on memory [m2s-2]
```

# Why do we use Ruby?

- Since we wanted a language for **daily** data analysis
  - Easy (fast) to write
  - Interactive use → like GrADS
  
- Python is also fine (but we love Ruby)

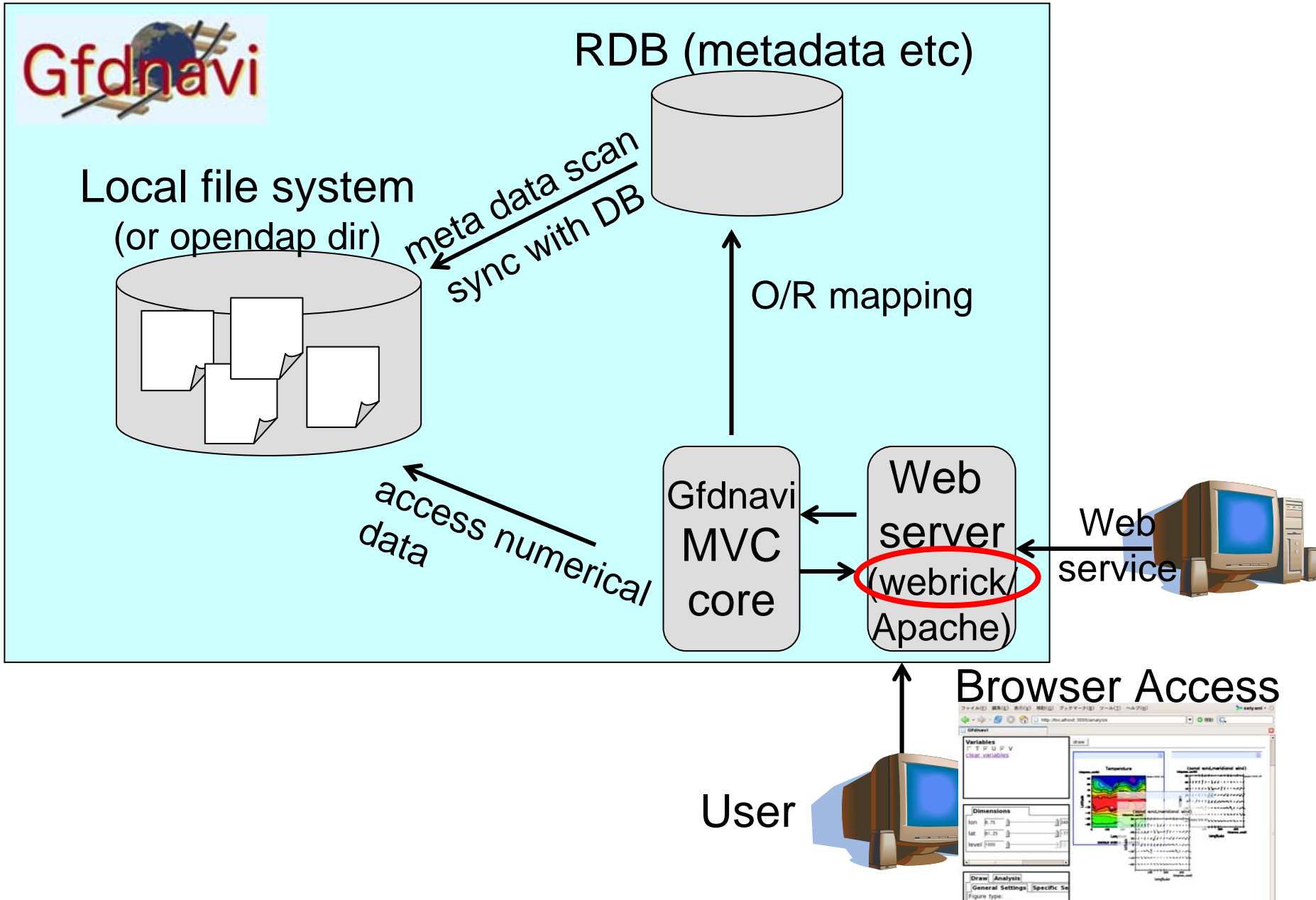
# Introducing Gfdnavi

*Since 2006*

Early History (Aug 2006):

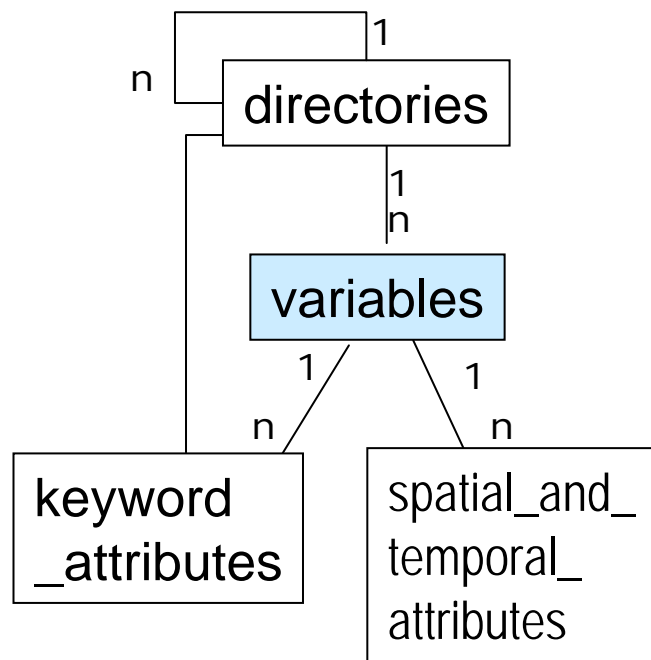
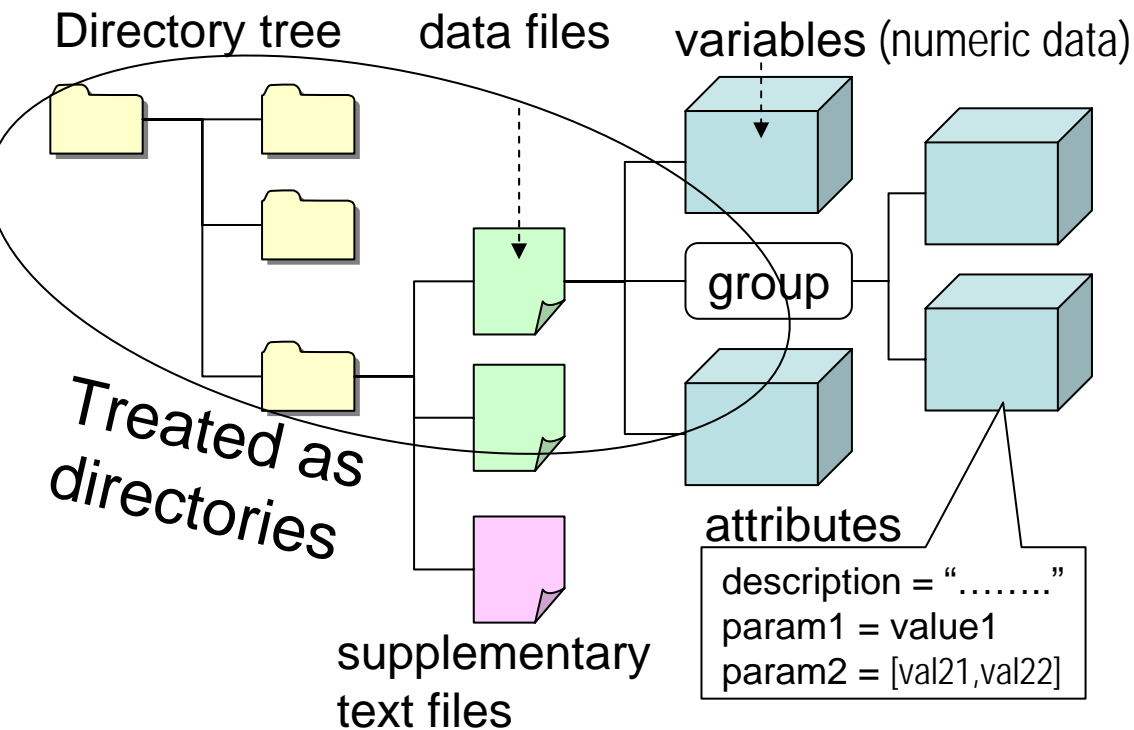
- ✓ Rough design by Horinouchi etc (at a meeting of the GFD-Dennou davis project)
- ✓ First implementation by **S Nishizawa** – In two weeks (since then he is the most contributing to its development)

# Overview



# Metadata DB

- Metadata
  - name-value attributes; with a few standard field names
  - geospatial- and time-coordinate info
  - size, user info etc
- Directory structure (inherit metadata from parent directories)
- Generated by automatic scan (with a command)
  - variables: reading attributes through GPhys
  - directories: directory name and “Readme”-type texts



# User Interface

**Home** : Independent simple html → replaceable










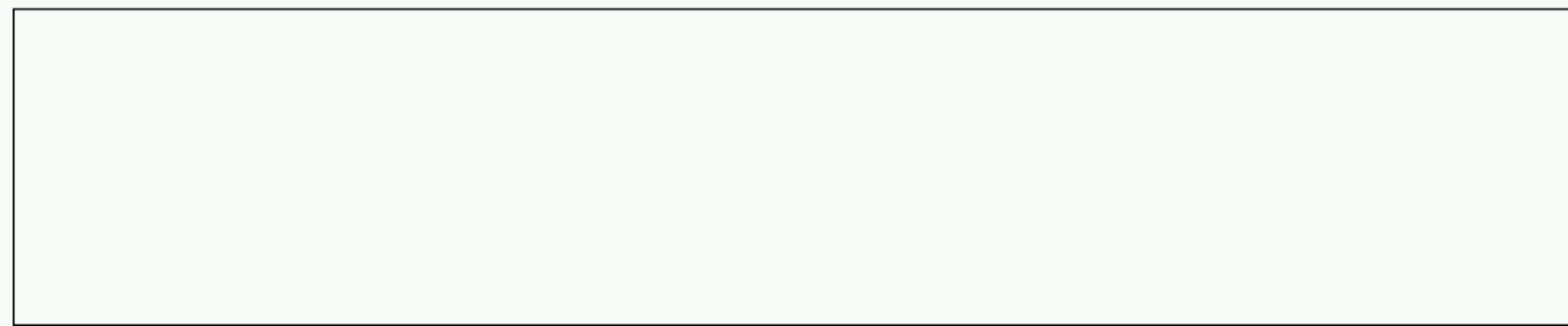
## GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)[Help](#)[Search \(Explorer\)](#) [Search \(old version\)](#)

Select from directory tree:

 [clear tree](#)

- localhost
  -   /
- localhost
  -   <http://test.opendap.org/opendap-3.7/>





# Functionality



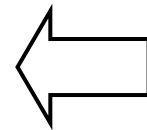
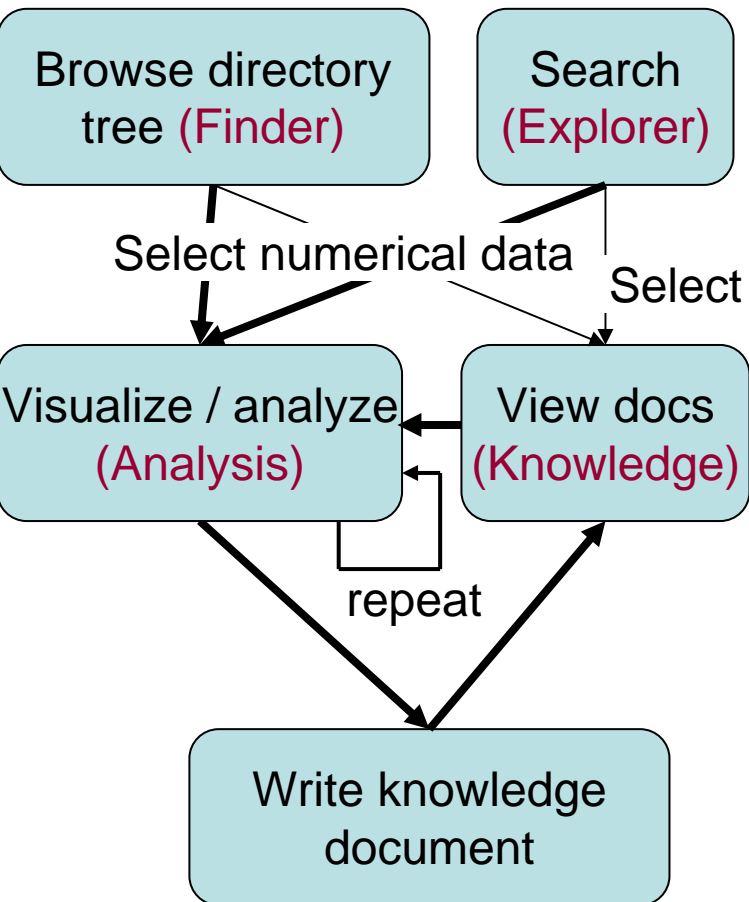
GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)

[Help](#)



Browser UI Header

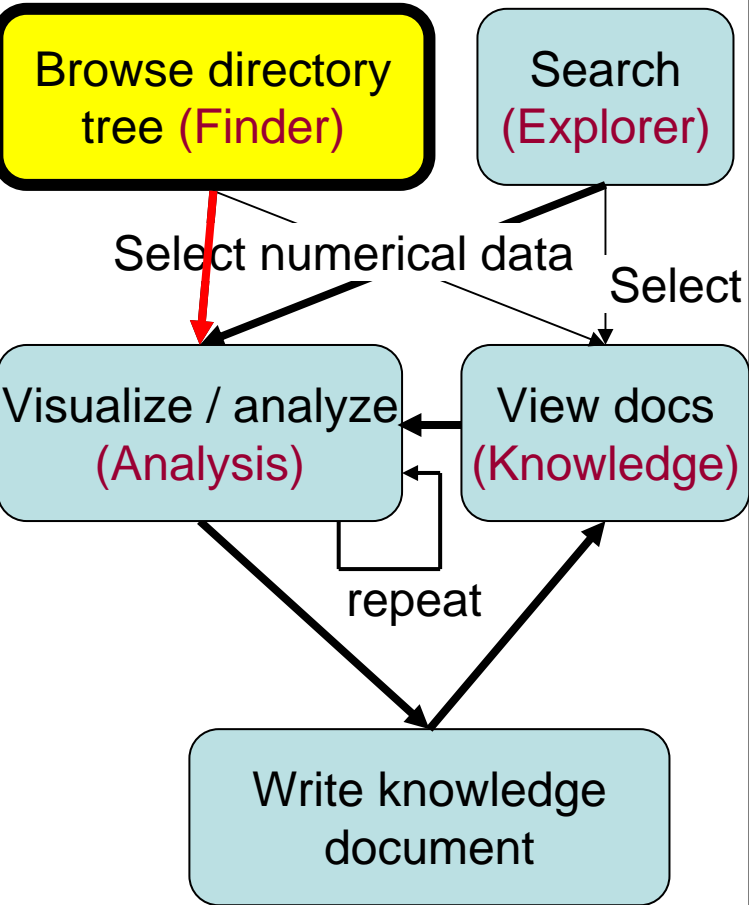


Typical work flow  
to use Gfdnavi's  
browser UI

# Functionality



[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Know](#)



The screenshot shows the Gfdnavi interface. On the left is an 'MS Explorer-like tree' showing a directory structure: / > samples > imadata > reanalysis > era40 > ncep > T.jan.nc. On the right is a file list for the selected file 'T.jan.nc'. The file list has columns for 'name' and 'title'. The 'name' column contains 'Anal/Viz', 'Details', and 'Show'. The 'title' column contains 'T.jan.nc', 'T.jan.zonal\_mean.nc', 'UV.jan.nc', and 'T.jan.100hPa.png'. A red box highlights the 'Anal/Viz' and 'Details' buttons in the file list, with a callout that says 'Select variables in this file to analyze / visualize'. Below the file list are buttons for 'Analyze/visualize checked items' and 'Show checked items'. A callout 'Further details (metadata)' points to the 'Details' button.

The screenshot shows the file details for 'T.jan.nc'. At the top, there are links for 'open node tree' and 'Download this file'. Below that, the file name 'T.jan.nc' is shown with its type '[plain file]' and its path '/samples/reanalysis/ncep/T.jan.nc'. Underneath is a section for 'Ancestors' with a numbered list: 1. /, 2. samples, 3. reanalysis, 4. ncep, 5. T.jan.nc. Below the ancestors list, there is a 'Description:' section with a numbered list: 1. /, 2. samples.

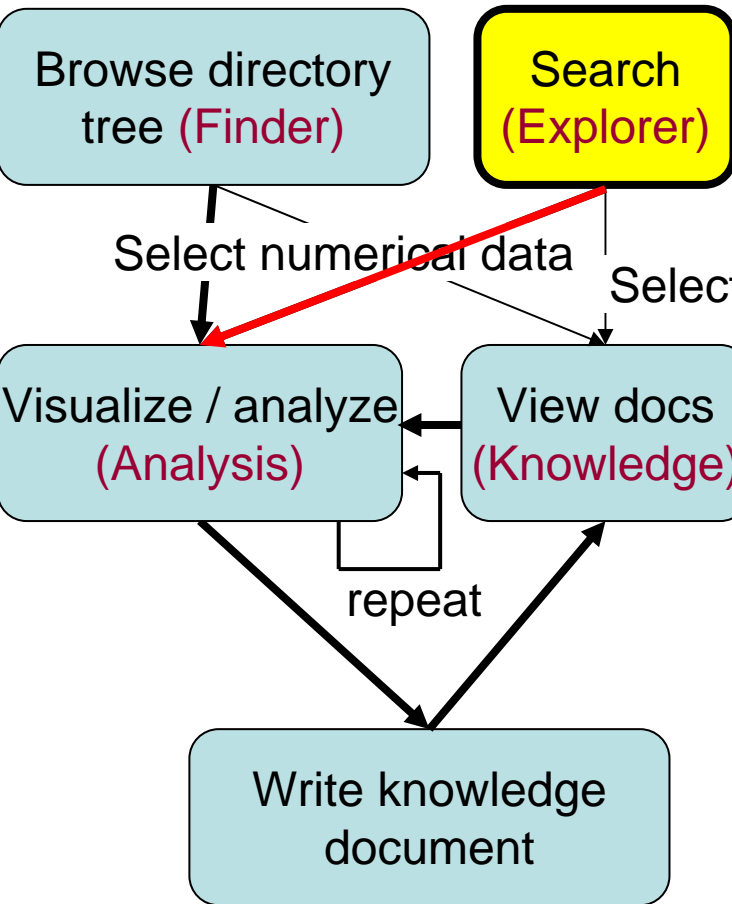
# Functionality



GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)

[Help](#)



The screenshot shows the GFDNAVI web interface. At the top, there is a navigation bar with links: [Top](#), [Finder](#), [Explorer](#), [Analysis](#), [Knowledge](#), [Login](#), and [Help](#). Below the navigation bar, there is a search area with a 'Query Conditions' section containing a checked box for '[F]temperatu' and a 'Free Keywords' section. A 'Free text' callout points to the search input field. Below the search area, there is a 'Keyword' section with a list of variables: 

- long\_name(62)
- units(62)
- standard\_name
- coordinates(30)
- Description(8)
- NOEP...
- ...
- least significar
- level\_desc(3)
- parent\_stat(3)
- precision(3)

An 'Attributes' callout points to this list. A 'Search with Google Maps' callout points to a map showing a world view with several red location pins. A yellow callout 'Select a variable to analyze / visualize' points to the 'Anal/Viz' button in the results table. The results table has columns for 'Anal/Viz' and 'Details'. A 'Results' callout points to the table header. The bottom of the page shows a 'Powered by Google' logo and a scale bar.

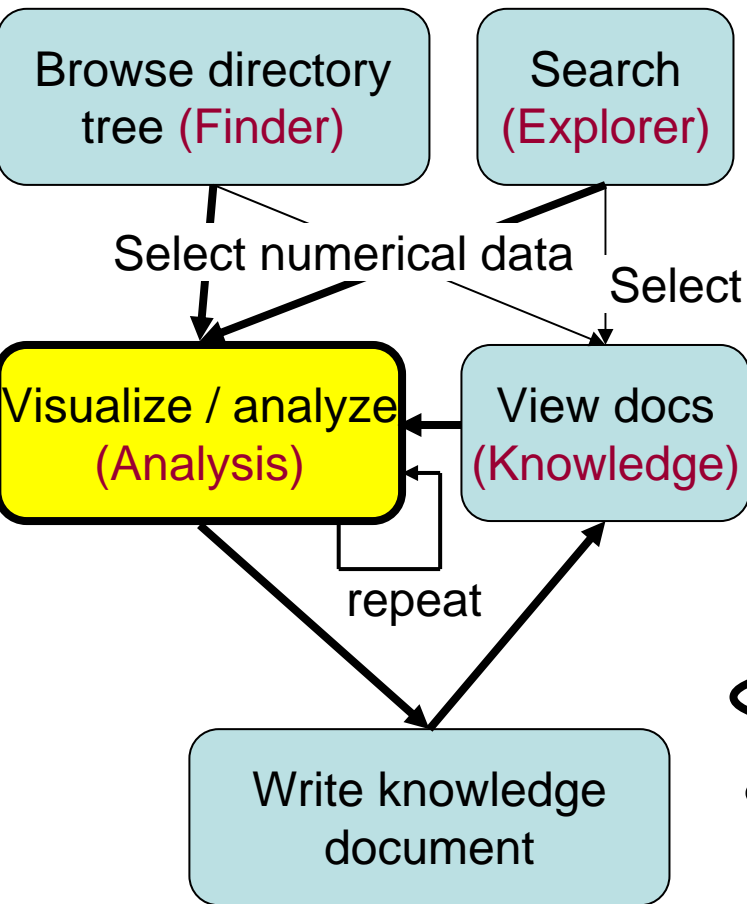
# Functionality



GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)

[Help](#)



The screenshot shows the GFDNAVI web interface. A 'Variables' window displays the path `/samples/reanalysis/era40/t.jan.nc/t` and attributes: `missing_value: 157.063151760226`, `long_name: Temperature`, and `units: K`. A 'Temperature' plot shows a polar projection map with contour lines. The 'Options' window has a 'draw!' button circled in red. Below it, 'Figure type' is set to 'tone', and 'Animation' is checked with 'levelist' as the dimension to animate. The 'Projection Type' is 'polar stereo projection'. Annotations include: 'Save in the DB (login needed)', 'Ruby Script & Minimum Subset Data' with links for 'download script and data', 'save diagram', and 'link to this diagram'; 'Get the URL to redraw the img'; and 'supply your own'.

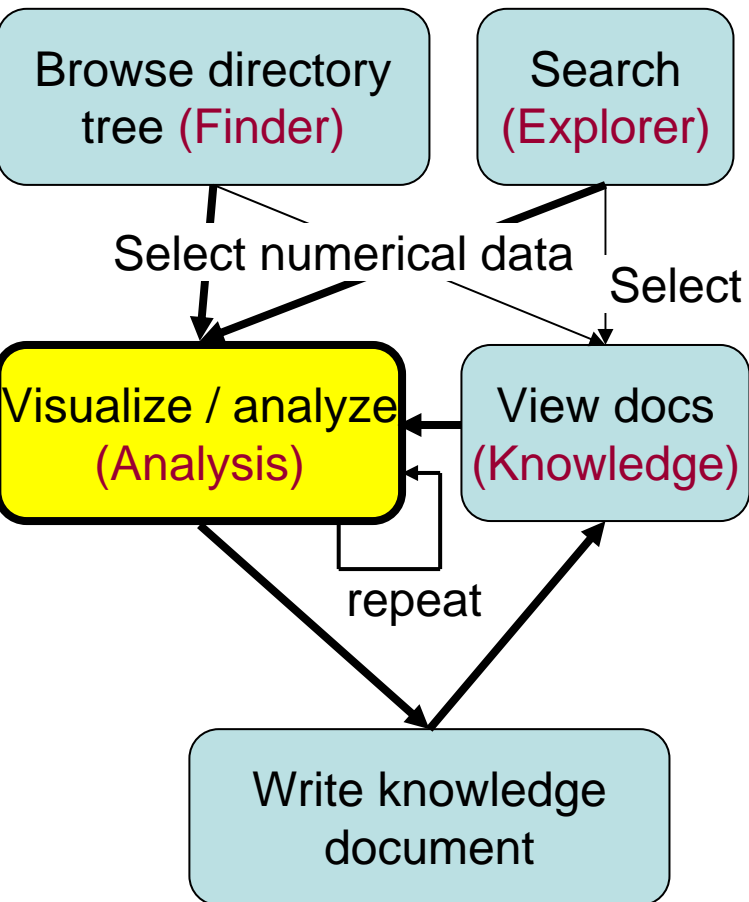
# Functionality



GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)

[Help](#)



Variables

U  V  t

[clear variables](#)

Axes

Dimensions

lon	8.75	348.0
lat	-77.5	81.2
level	10	100

[map](#)

Options

Draw ? Analysis ?

**analyze!**

Function: cut

- cut
- mean
- stddev
- addition
- division
- multiplication
- subtraction

[add function](#)

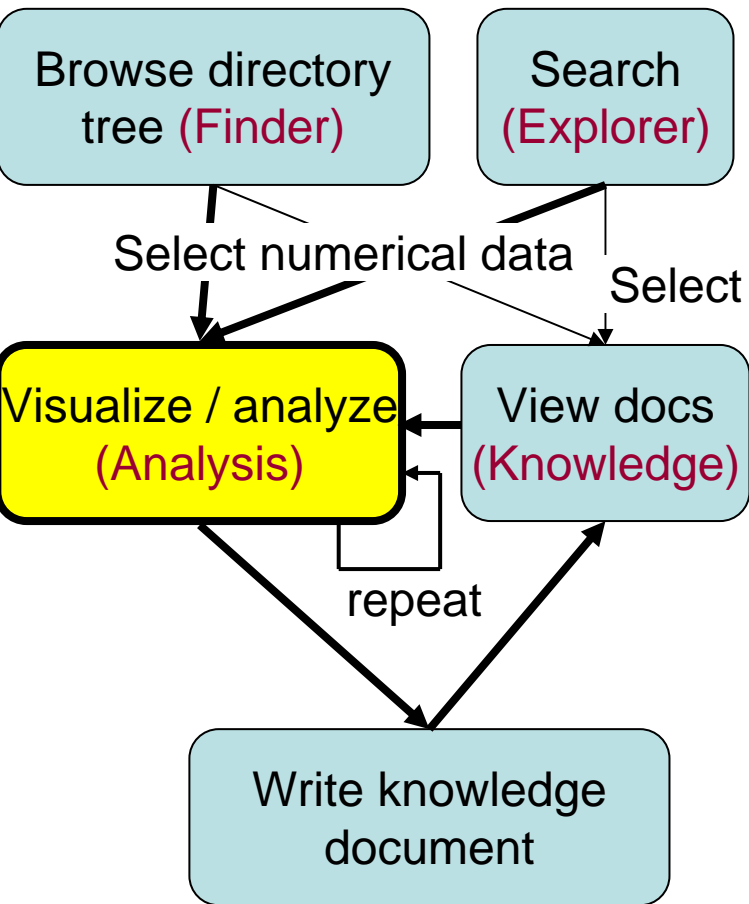
# Functionality



GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)

[Help](#)



The screenshot shows the GFDNAVI interface with three main panels:

- Variables:** Contains checkboxes for 'U', 'V', and 't'. 'U' and 'V' are checked. A 'clear variables' link is present below.
- Axes:** Features a 'Dimensions' section with three rows of sliders:
  - lon: 8.75 (range 348)
  - lat: -77.5 (range 81.2)
  - level: 10 (range 100)A 'map' link is located to the left of the sliders.
- Options:** Has two tabs: 'Draw' and 'Analysis'. The 'Analysis' tab is active, showing a red-bordered button labeled 'analyze!'. Below it, a 'Function' dropdown is set to 'mean', with an 'add function' link. A list of dimensions (lon, lat, level) is shown with arrows indicating they are selected for the 'mean' function.

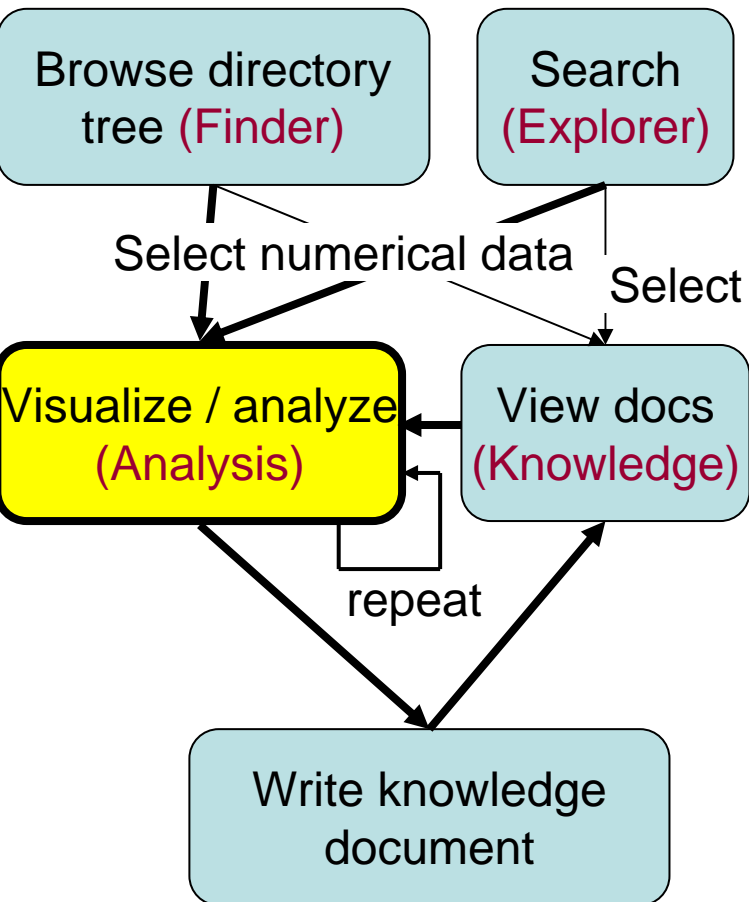
# Functionality



GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)

[Help](#)

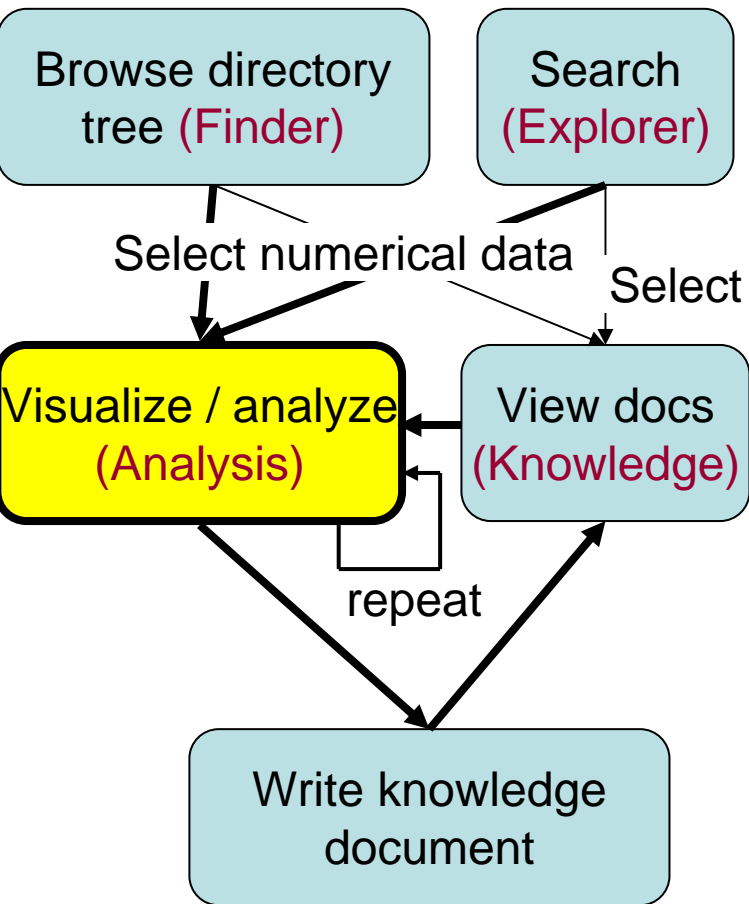


The screenshot shows the login page of GFDNAVI. At the top, there is a navigation bar with links: [Top](#), [Finder](#), [Explorer](#), [Analysis](#). Below the navigation bar, the text 'Please login' is displayed. There are two input fields: 'Login:' and 'Password:'. Below the input fields, there are two buttons: 'login' and 'signup'.

# Functionality



[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowl](#)



## Create Function

### Function

name	<input type="text" value="spectrum"/>
save directory	<input type="text" value="/usr/root/functions"/>
description	<input type="text" value=" FFT ^2 along a specified dimension"/>
group	<input type="text" value="only me"/> <input type="text" value="--groups--"/> <a href="#">create_group</a>
number of input variables	<input type="text" value="1"/>
number of arguments	<input type="text" value="1"/>
script	<pre>{arg0, gphys0} [ gphys0.fft(arg0).abs ** 2 ] }</pre>
number of output variables	<input type="text" value="1"/>

### Function Arguments



# Functionality



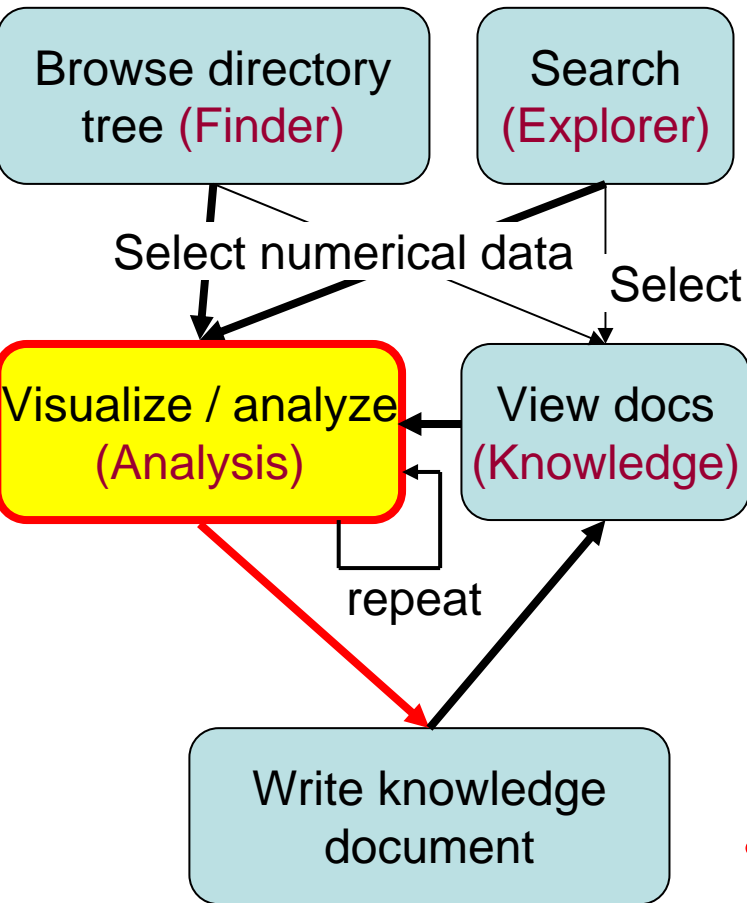
[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Know](#)



GFDNAVI

[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [User](#) [Logout](#)

[Help](#)



**Variables**

t

[clear variables](#)

upload

**Axes**

**Dimensions**

longitude 0

latitude 90

levelist 1

map

**Options**

Draw ? Analysis ?

**draw!**

**General Settings** **Specific Settings**

Record visualization for statistics

Figure type: tone

the 1st Dim: longitude

the 2nd Dim: latitude

Animation

dimension to animate: levelist

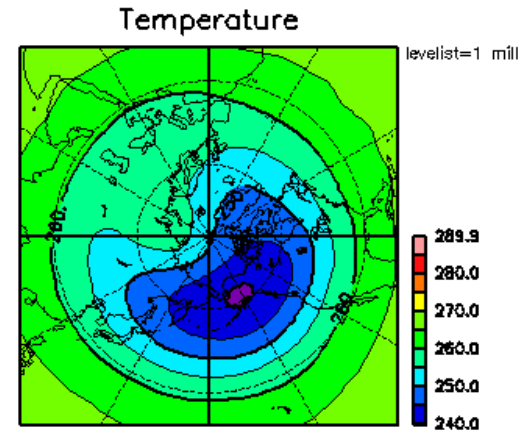
Projection Type: polar stereo projection

Pile up

Keep diagrams

Diagram size: large med small x-small

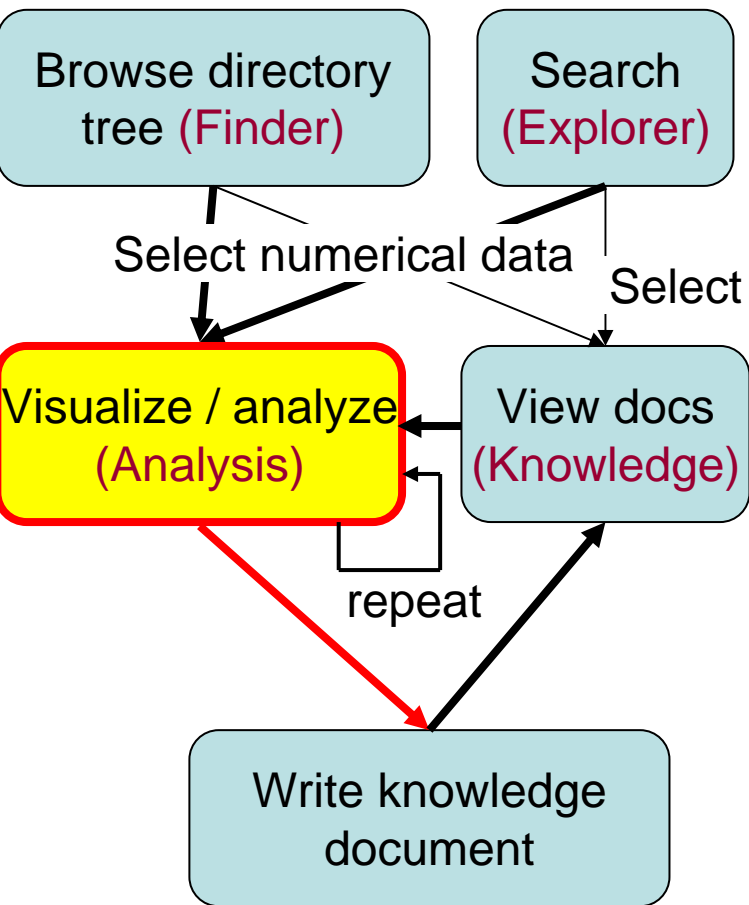
Viewport vxmin, vxmax, vymin, vymax (0 to 1):



Create a Knowledge Document with this/these Image(s)



Top Finder Explorer Analysis Know



F

clear\_variables

upload

### Axes

Dimensions

longitude 0

latitude 90

levelist 100

### Options

Draw Analysis

**draw!**

General Settings Specific Settings

Record visualization for statistics

Figure type: tone

the 1st Dim: longitude

the 2nd Dim: latitude

Animation

dimension to animate: levelist

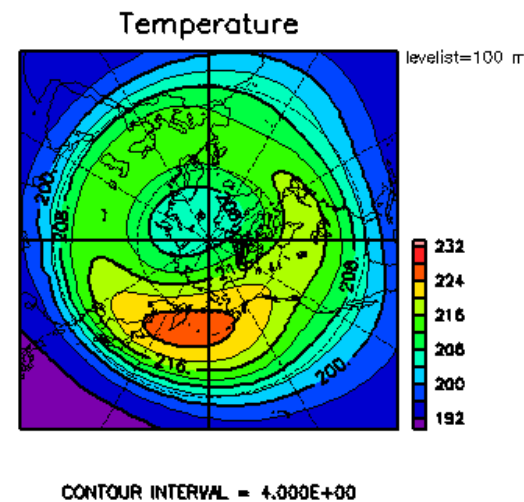
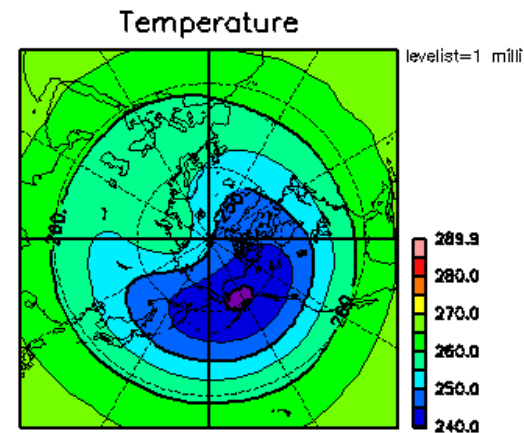
Projection Type: polar stereo projection

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



Create a Knowledge Document with this/these Image(s)

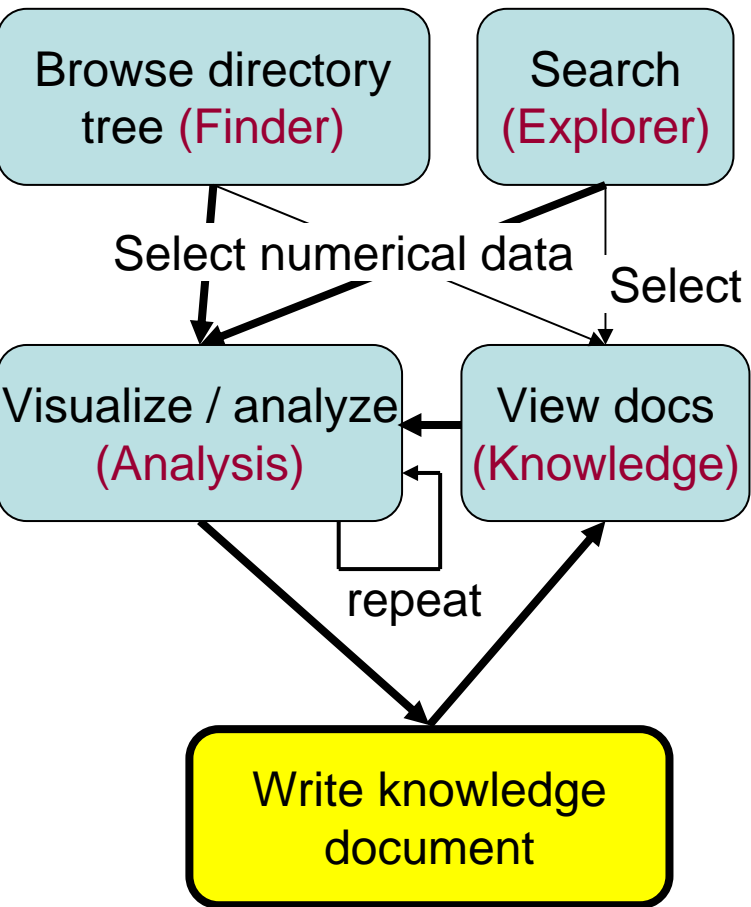
# Functionality



GFDNAVI

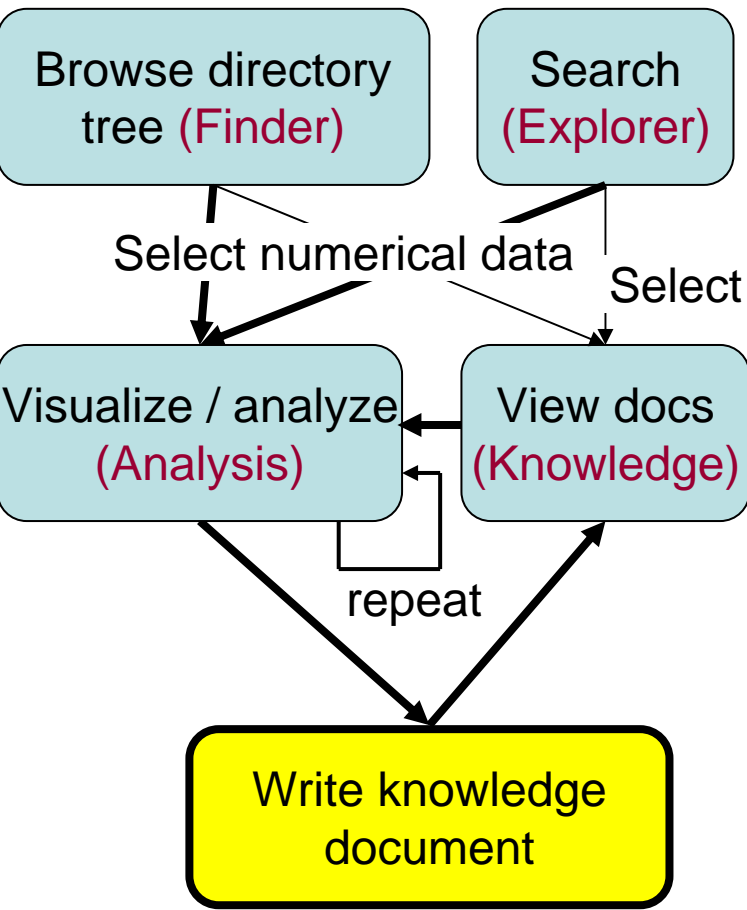
[Top](#) [Finder](#) [Explorer](#) [Analysis](#) [Knowledge](#) [Login](#)

[Help](#)



The screenshot shows the login page of GFDNAVI. At the top, there is a blue navigation bar with links: [Top](#), [Finder](#), [Explorer](#), and [Analysis](#). Below the navigation bar, the text 'Please login' is displayed. There are two input fields: 'Login:' and 'Password:'. Below the input fields, there are two buttons: 'login' and 'signup'.

# Full Save Images and Create a New Knowledge



Title:  Author:

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 storm track.

Path: /usr/root/knowledge/tmp/eraT.knlge  
e\ /usr/root/knowledge/folder1/folder2/writing.knlge


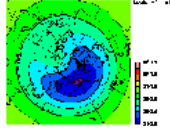
visible to:  everyone

Choose a default layout :  size of figure:   %

input the number of figures in a row

Figure 1

Caption:

Temperature

File Name:

[view this image in the original size](#)

Figure 2

Caption:

Temperature

File Name:

[view this image in the original size](#)

[More Figure](#)

Layout :  size of figure:   %

input the number of figures in a row

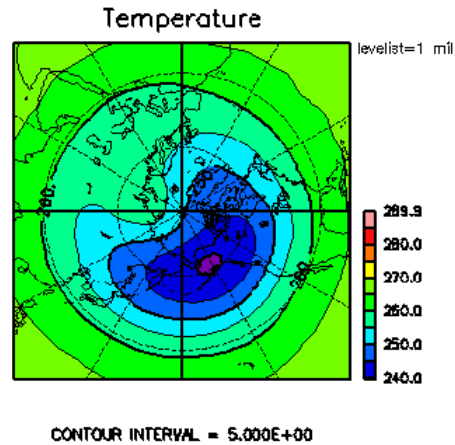
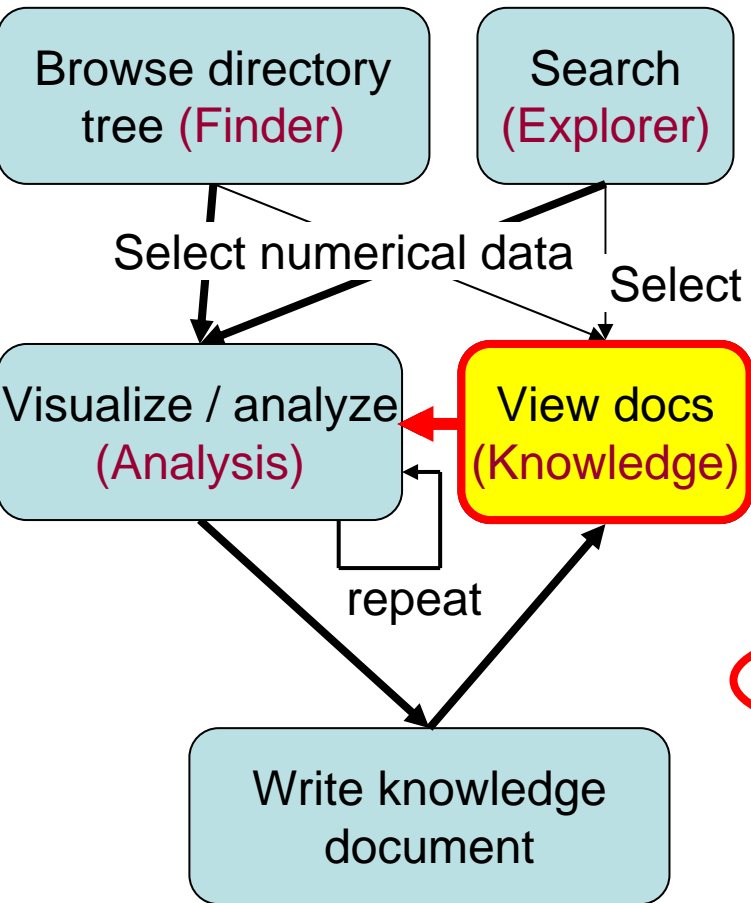


## ECMWF Reanalysis January Climatology

Author: T Horinouchi

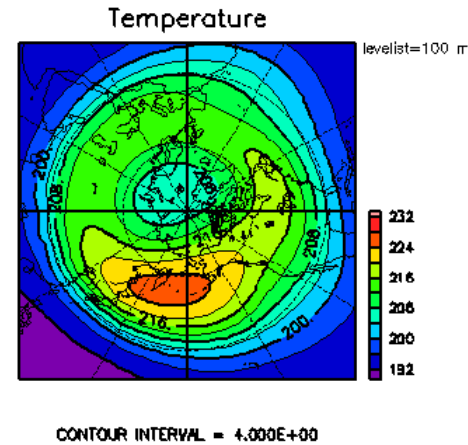
[Figure 1](#) shows the climatological temperature at 1 hPa using the ECMWF Reanalysis (ERA40) in the northern hemisphere. It shows that the climatological polar vortex is shifted to the Pacific side.

[Figure 2](#) shows is the same as Fig1 but for 100 hPa. It shows that the westerly jet is strong in the Pacific storm track.



[<redraw this image>](#) [<Get the URL>](#)

Fig. 1. ERA Jan T at 1 hPa



[<redraw this image>](#) [<Get the URL>](#)

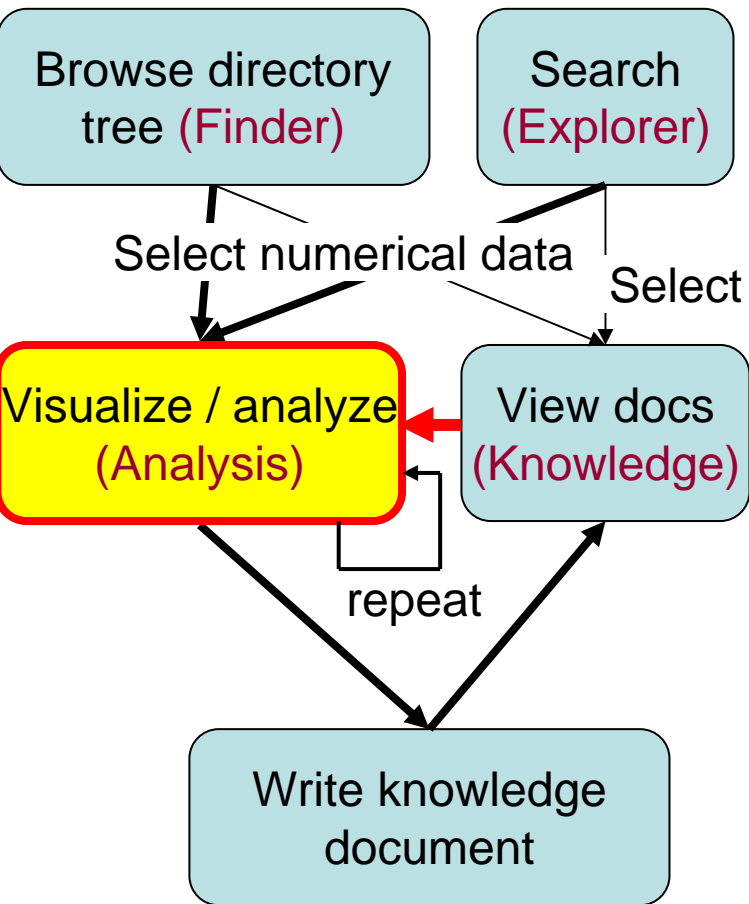
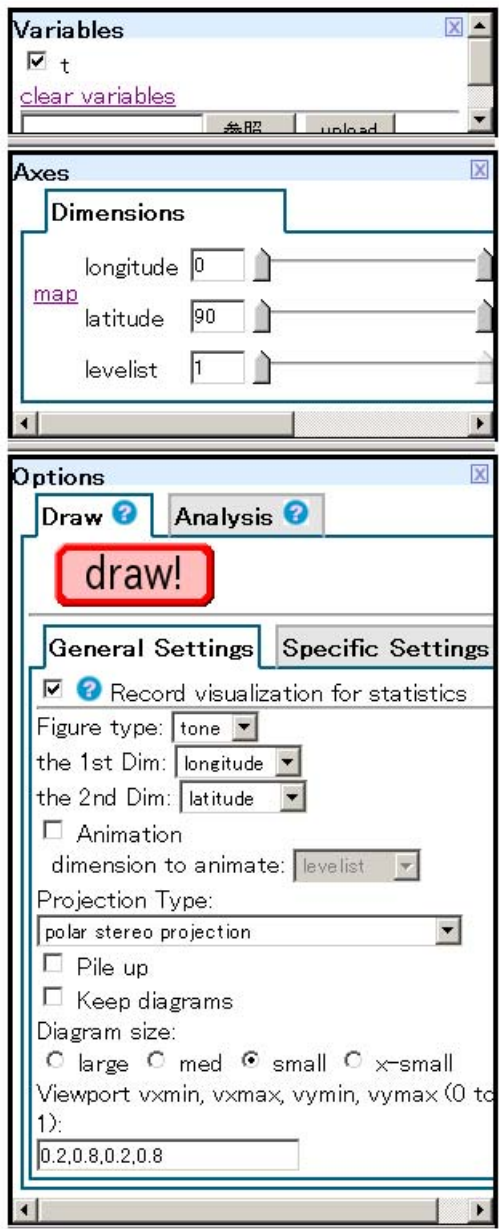
Fig. 2. ERA Jan T at 100 hPa

Path: /usr/root/knowledge/tmp/era.T.knlge

[Edit](#) | [Back to List](#)

There are no comment on this document.

[Back to List](#)



## Listing Knowledge Documents

New Knowledge

Previous **1** 2 Next

[Temperature data from era40](#) by root last update: Sun Sep 14 13:02:18 [show full](#)  
 about figure. temperature data. levelist 1, so height is about 48km.  
 path: /usr/root/knowledge/temperature\_data\_from\_era40.knlge [Edit](#) [Delete](#)

[Typhoon Information](#) by root last update: Sun Sep 14 12:54:21 [show full](#)  
 A typhoon occurred at east of philippine A typhoon 5 occurred on July 2005. The figure means amount of  
 path: /usr/root/knowledge/typhoon4.knlge [Edit](#) [Delete](#)

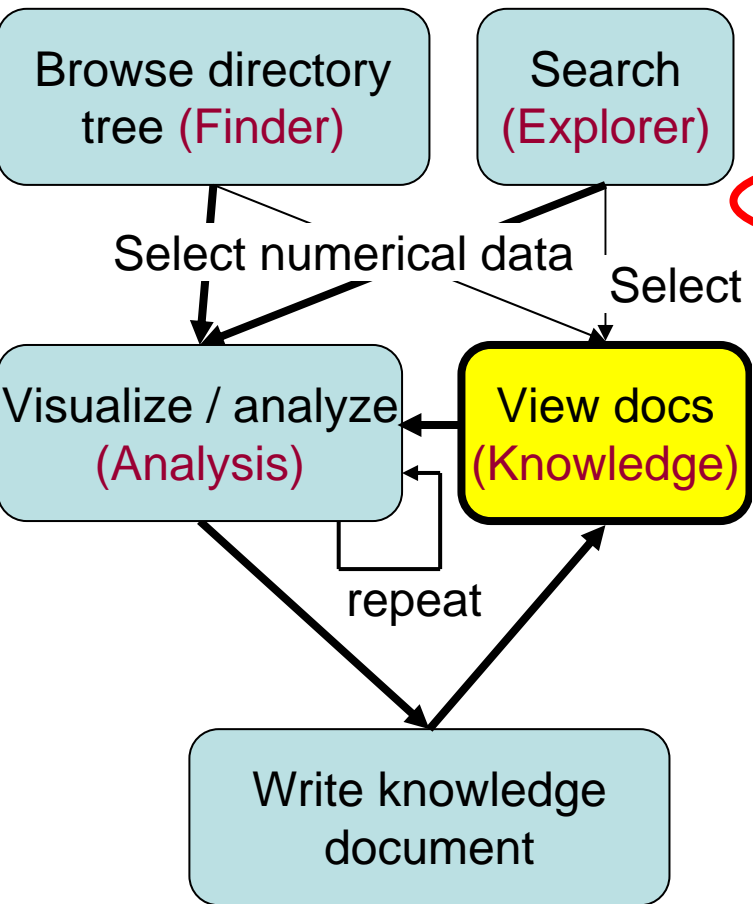
[Typhoon Information](#) by root last update: Sun Sep 14 12:23:16 [show full](#)  
 A typhoon occurred at east of philippine A typhoon 5 occurred on July 2005. The figure means amount of  
 path: /usr/root/knowledge/typhoon3.knlge [Edit](#) [Delete](#)

[台風情報](#) by root last update: Mon Sep 08 01:55:36 [show full text here](#) [display comment](#)  
 台風発生 2005年7月、台風5号が発生しました。図は2005年7月16日の、1時間当たりの降雨量を表してい  
 path: /usr/root/knowledge/typhoon.knlge [Edit](#) [Delete](#)

[複数の絵を描画する方法](#) by root last update: Sat Aug 23 21:46:15 [show full](#)  
 はじめにこの文書では、多くの図が入った知見文書を作成するために、複数の絵を描く方法について解説  
 に関してはKnowledge 機能の使い方/samples/how\_to\_knowledge.knlgeをご覧ください。...  
 path: /samples/how\_to\_draw\_multiple\_images.knlge [Edit](#) [Delete](#)

Previous **1** 2 Next

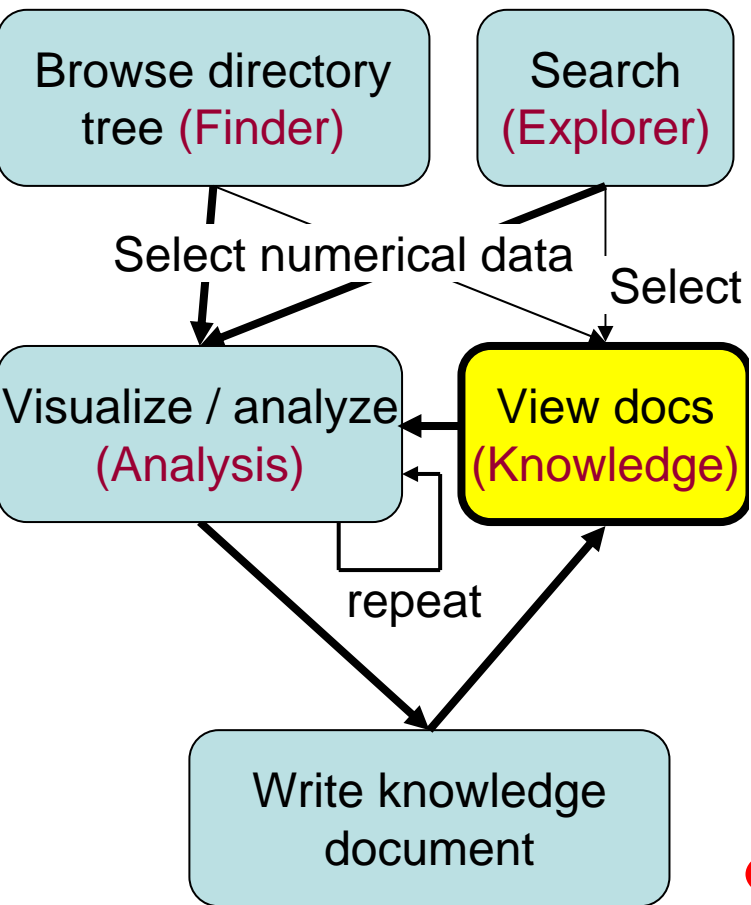
New Knowledge







Top Finder Explorer Analysis Know



# Typhoon Information

Author: Akinori

## A typhoon occurred at east of philippine

A typhoon 5 occurred on July 2005.

The figure means amount of rainfall per hour at July 16, 2005. We can see a typhoon east of philippine.

## Forecast of Course of typhoon

According to the forecast of Japan Meteorological Agency, typhoon 5 will change direction of movement near Taiwan. The sea around Okinawa is warmed by the intense heat of days, so it seems that the typhoon will move further. There is possibility of coming off from the expectation and landing on West Japan. Please note this in the future.

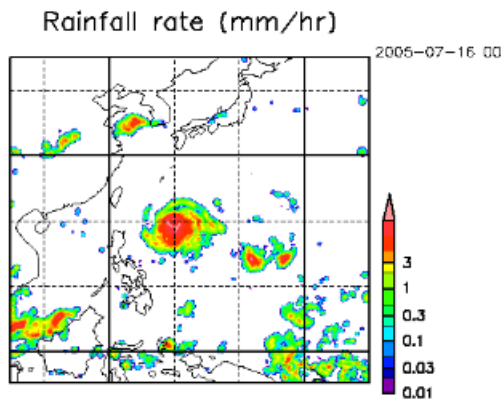


Fig 1. Rainfall

Path: /usr/root/knowledge/typhoon3.knlge

[Edit](#) | [Back to List](#)

There are no comment on this document.

[Back to List](#)



F

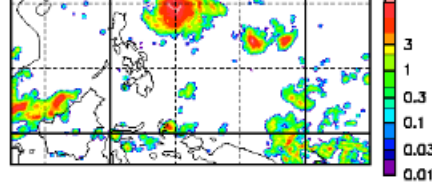
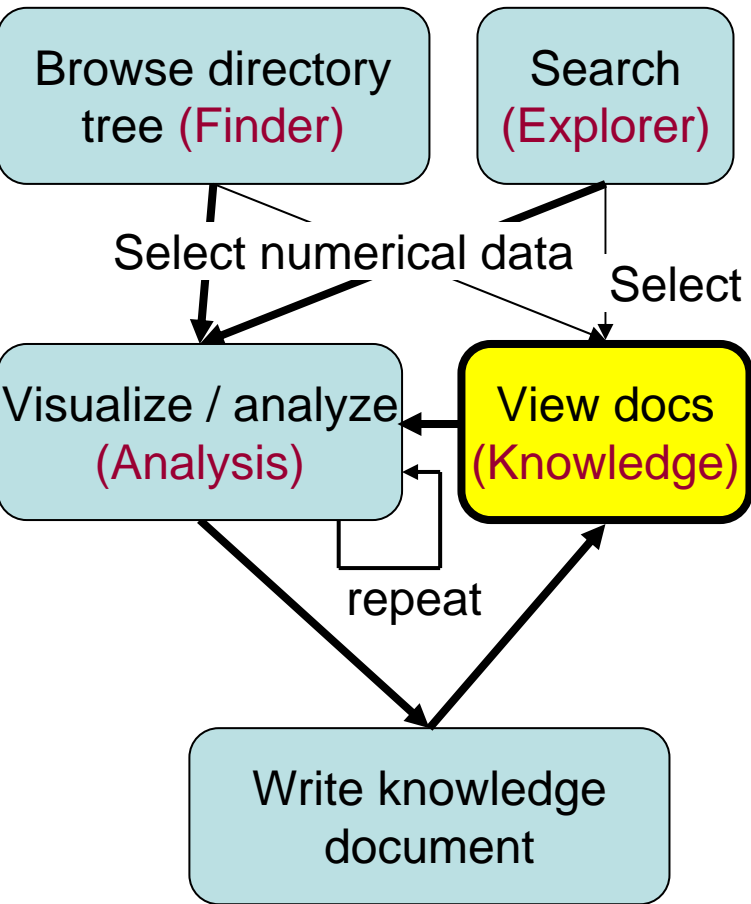


Fig. 1. Rainfall

Path: /usr/root/knowledge/typhoon4.knlge  
[Edit](#) | [Back to List](#)



[There are no comment on this document.](#)

[Add a Comment on this document.](#)

Title:  Author:

Textbody:

Choose a default layout :  size of figure:   %

input the number of figures in a row

Figure 1

Caption:

[delete](#)

Figure Path:

# F A typhoon occurred at east of philippine

A typhoon 5 occurred on July 2005.

The figure means amount of rainfall per hour at July 16, 2005. We can see a typhoon east of philippine

## Forecast of Course of typhoon

According to the forecast of Japan Meteorological Agency, typhoon 5 will change direction of movement near Taiwan. The sea around Okinawa is warmed by the intense heat of days, so it seems that the typhoon will move further. There is possibility of coming off from the expectation and landing on West Japan. Please note this in the future.

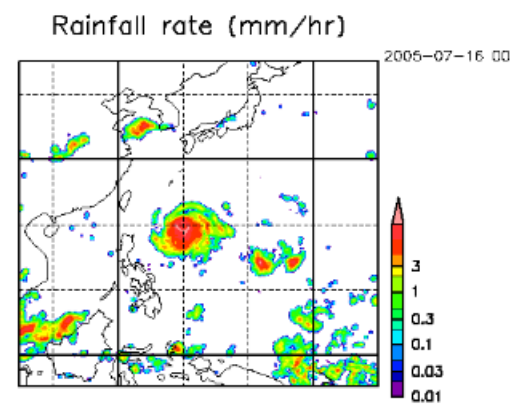
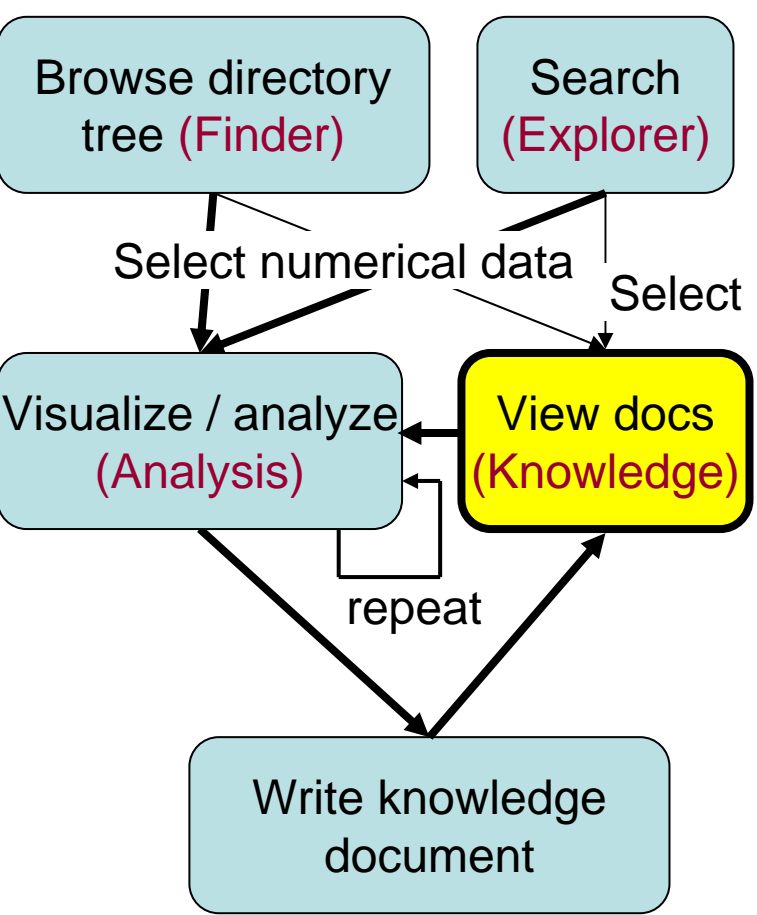
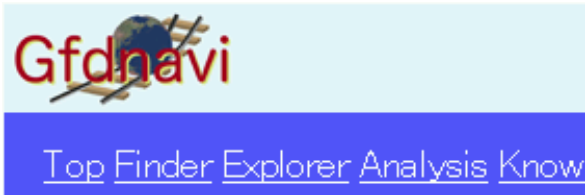


Fig. 1. Rainfall

Path: /usr/root/knowledge/typhoon3.knlge

[Edit](#) | [Back to List](#)

1 comment exists.

[Re\[1\]:Typhoon Information](#) author: **Akinori Tomobayashi** by **root** last update: **Sun Sep**

After all, typhoon 5 went for Taiwan and landed China.

[Back to List](#)

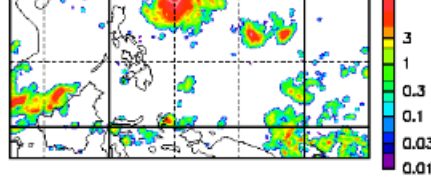


Fig 1. Rainfall

Path: /usr/root/knowledge/typhoon3.knlge

[Edit](#) | [Back to List](#)

1 comment exists.

Show Summary of Comments Hide Comments **Show full text of Comments**

[Re\[1\]:Typhoon Information](#) author: **Akinori Tomobayashi** by root last update: Sun Sep

After all, typhoon 5 went for Taiwan and landed China.

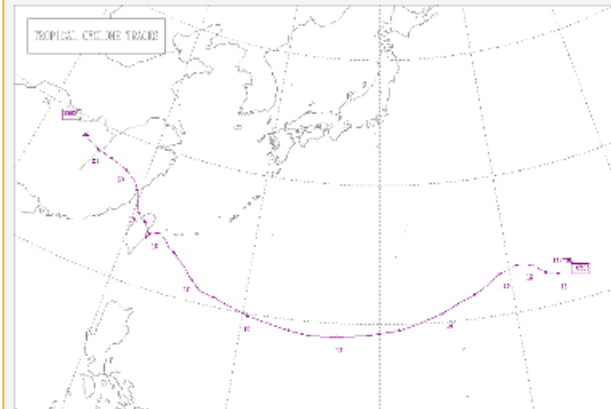
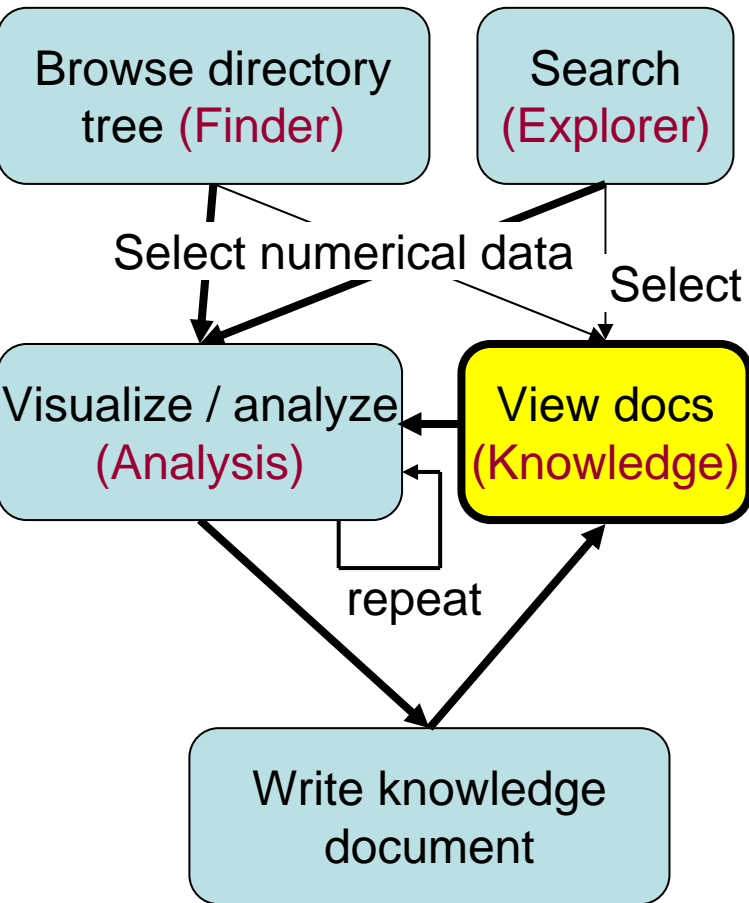


Fig 1. course of typhoon 5 in 2005.

Path: /usr/root/knowledge/typhoon3\_comment\_1.knlge

Write a Coment on this document.

[Back to List](#)



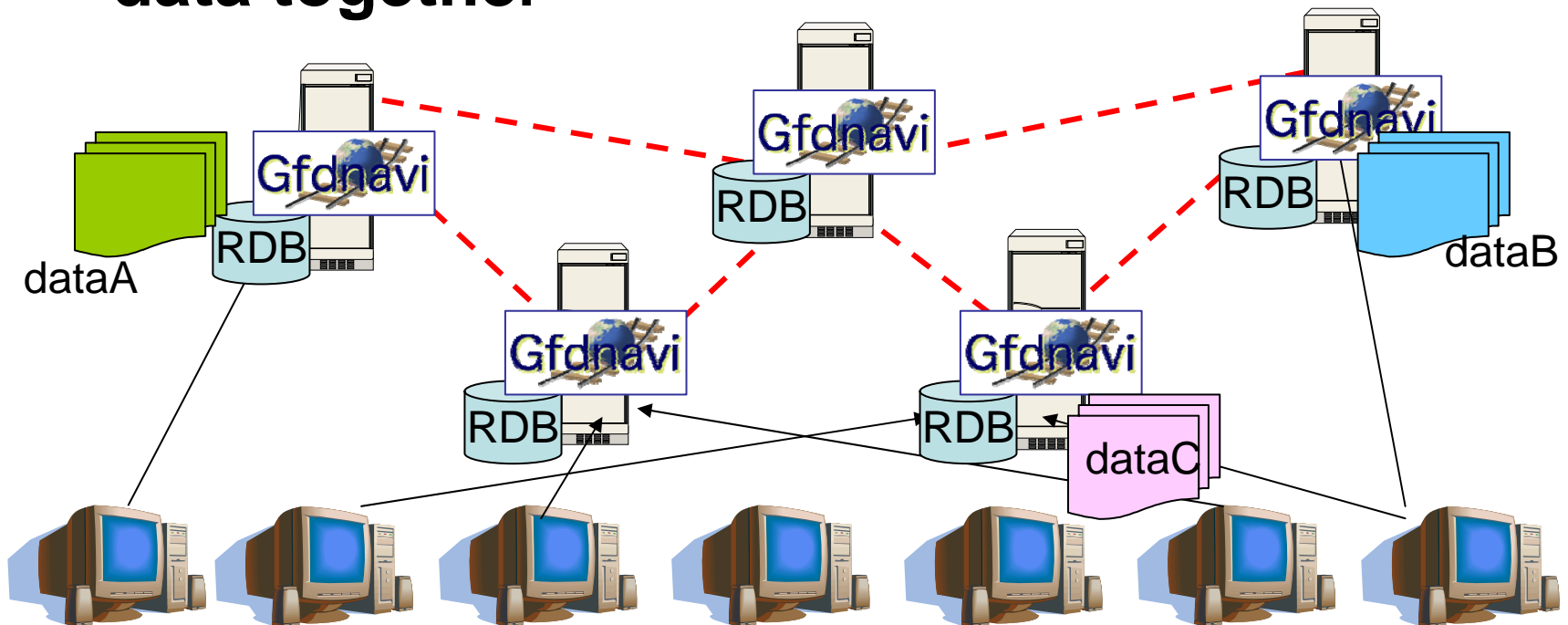
# Web service

→ Tomorrow by Seiya Nishizawa

# Network of Gfdnavi

Under development by C Watanabe (Ochanomizu Univ)

- ✓ To create peer-to-peer network for cross search and cross use **among Gfdnavi servers**
- ✓ Then one can access **local data and remote data together**



# Summary

- Novel features of Gfdnavi
    - Seamless coverage from desktop use to public data service (by having custom web server)
    - Programmability (on browser & by web service)
    - Documentation of analysis results (dynamically reproducible/extendible) (→ memos / reports / PR / **Blog for scientific collaboration**)
  - Good implementation
    - Extendibility (by using GPhys)
    - Swift development (by using RonR)
- Tomorrow by  
S Nishizawa

# Future Outlook

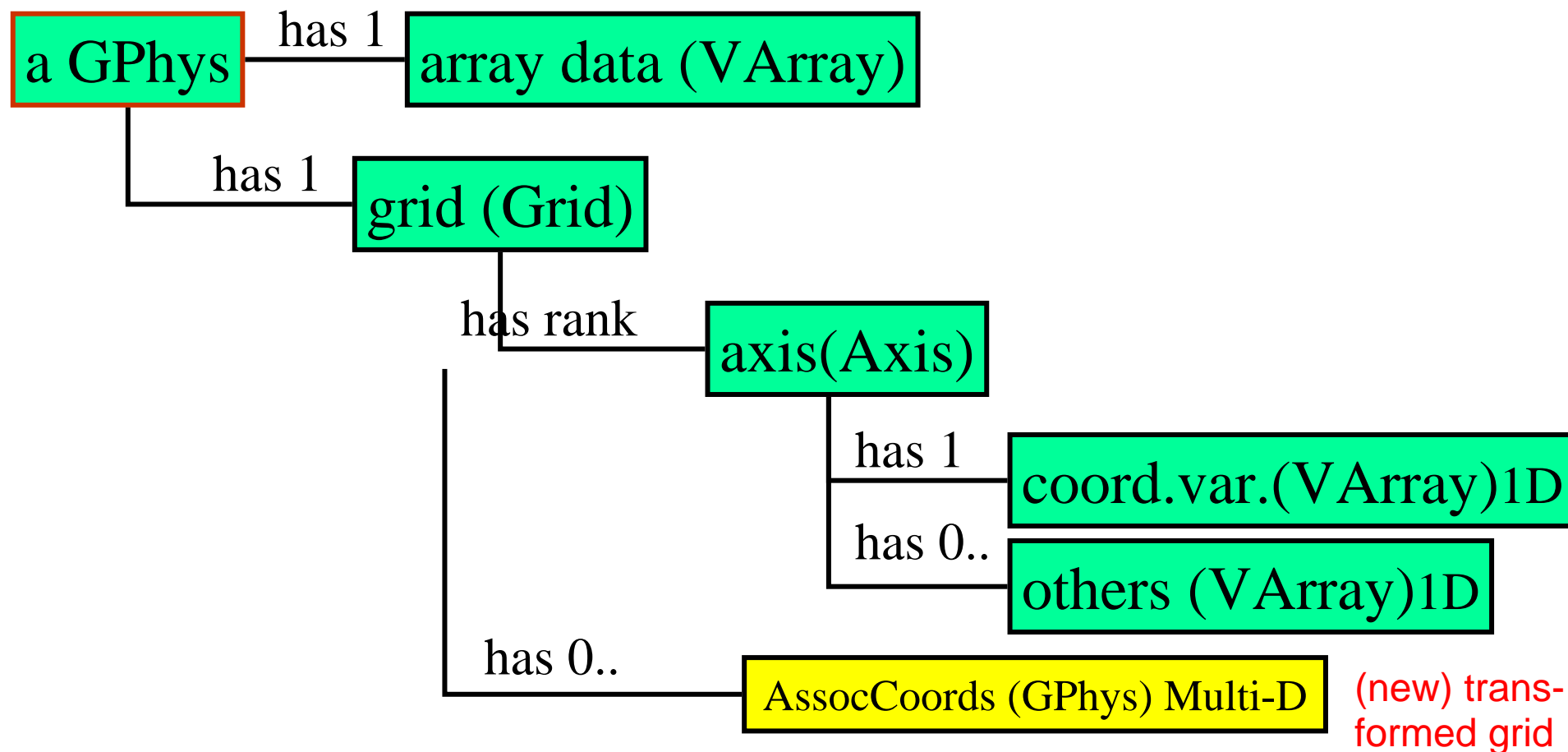
- Support Networking → **Create a Web of scientific data & knowledge**
- Increase analysis & visualization functionality (**many needed**)
- Improve remote API accesses (tomorrow's topic)

fin



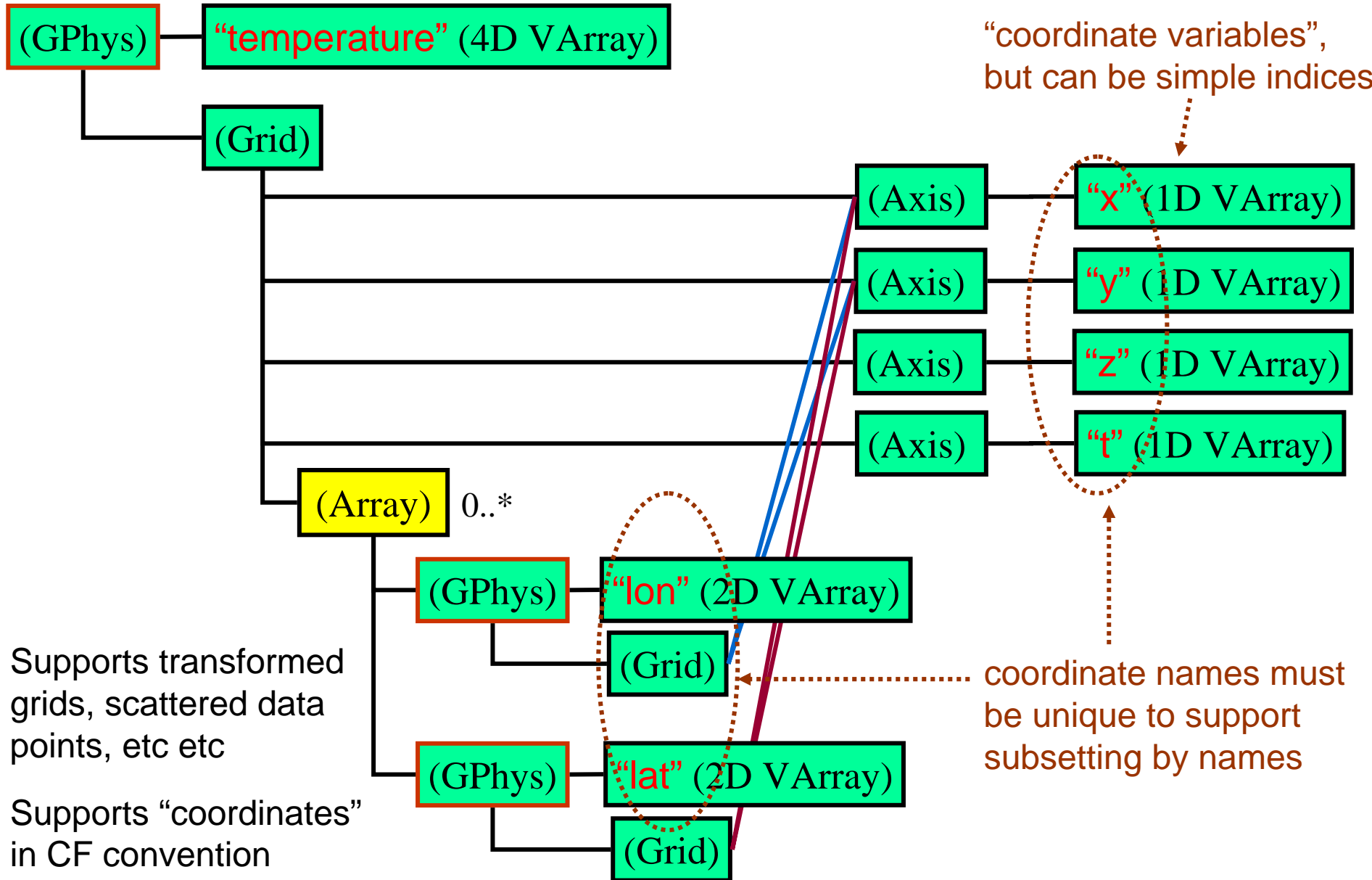


# GPhys (Gridded Physical quantity)



**VArray (Virtual Array)** – Abstracts Data Storage  
(Can be in file(s) or multi-D Array on memory; can also be a subset or aggregation of (an)other VArray(s))

# Example of GPhys's associated coordinates





# What is Ruby on Rails

<http://www.rubyonrails.org/>

Agile Web  
Development  
with Rails



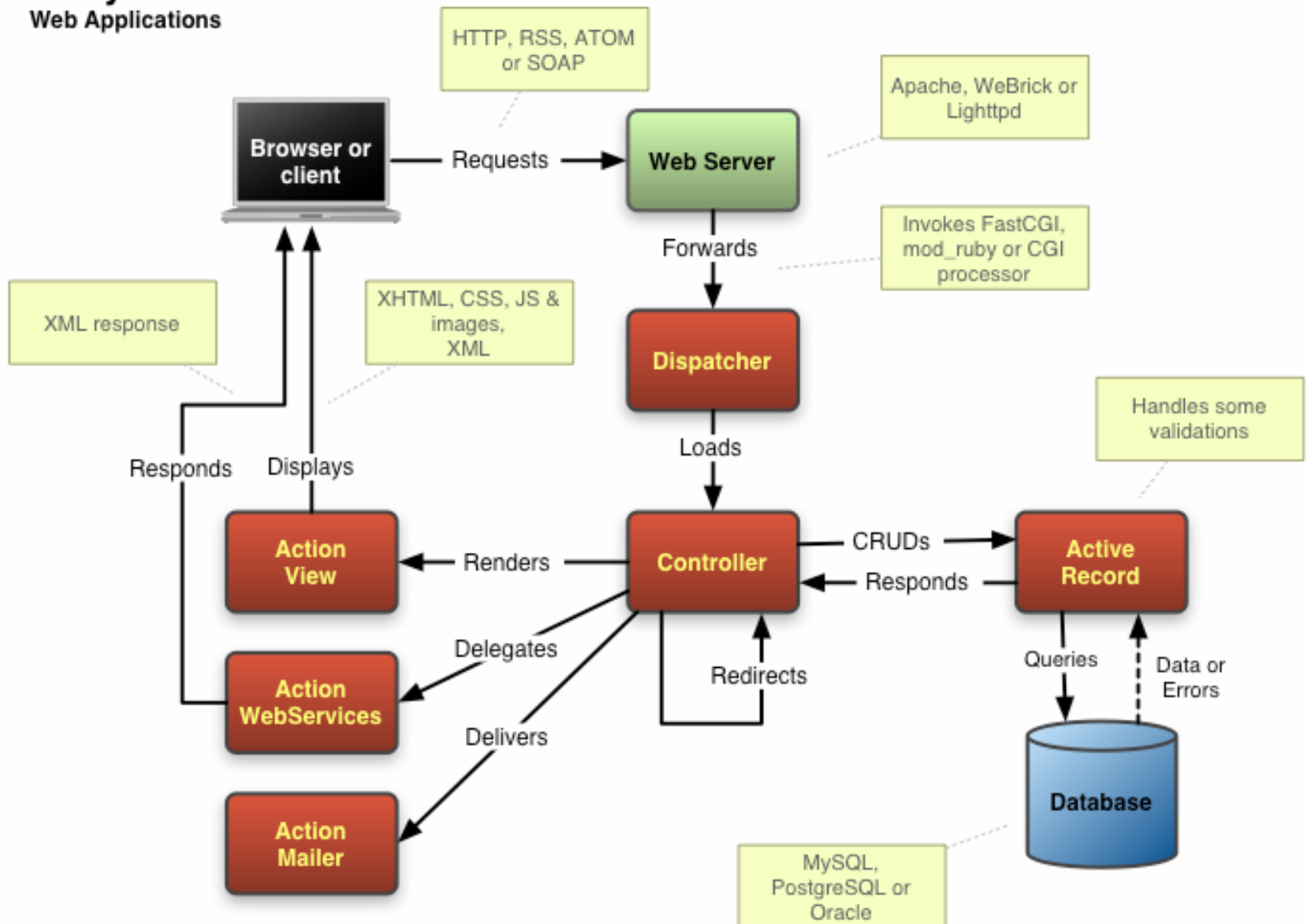
- Web development framework in Ruby
- With RDBMS (Mysql, Postgres, SQL Server, SQLite etc)
- Strong prototyping (e.g. Model-View-Controller (MVC) structure)
- Comprehensive library (covering Ajax and Web service)
- Ruby-embedded html
  - suitable to use our Ruby library
- Has a private web server (Webrick); also runs on Apache, lighttpd etc
  - One can personally run a web server anywhere with arbitrary port

# From “Understanding Rails MVC”:

<http://wiki.rubyonrails.org/rails/pages/UnderstandingRailsMVC>

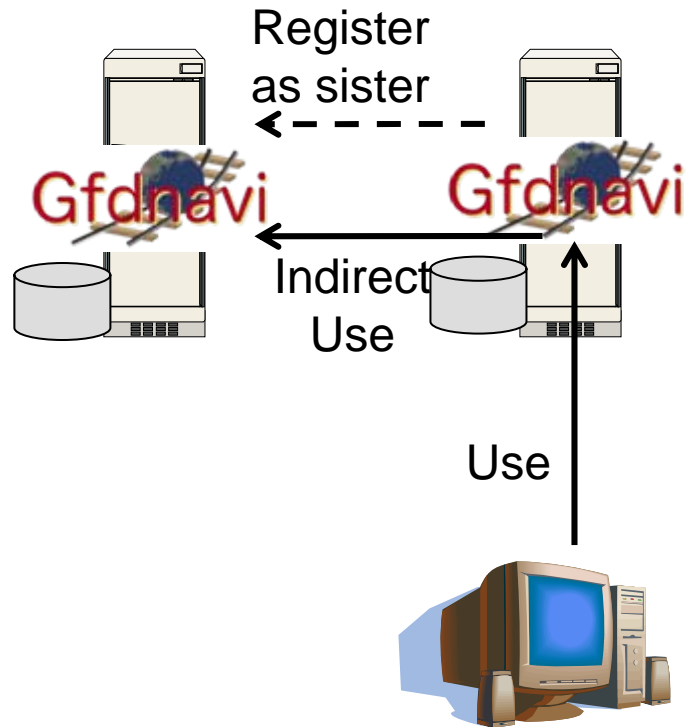
## Ruby on Rails

Web Applications

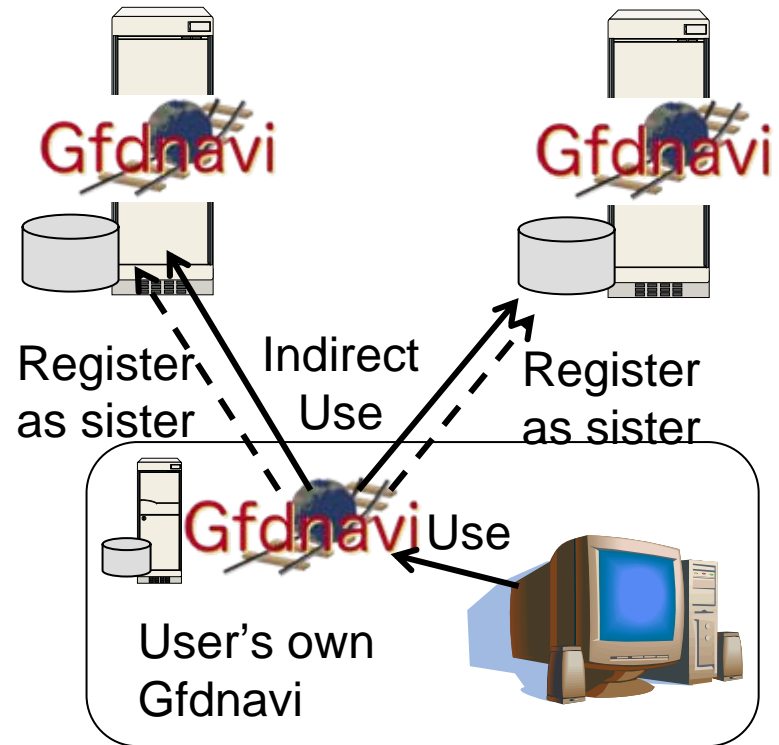


# Sister-server method

(a) Basic case: available in LAS.  
User can't choose peers

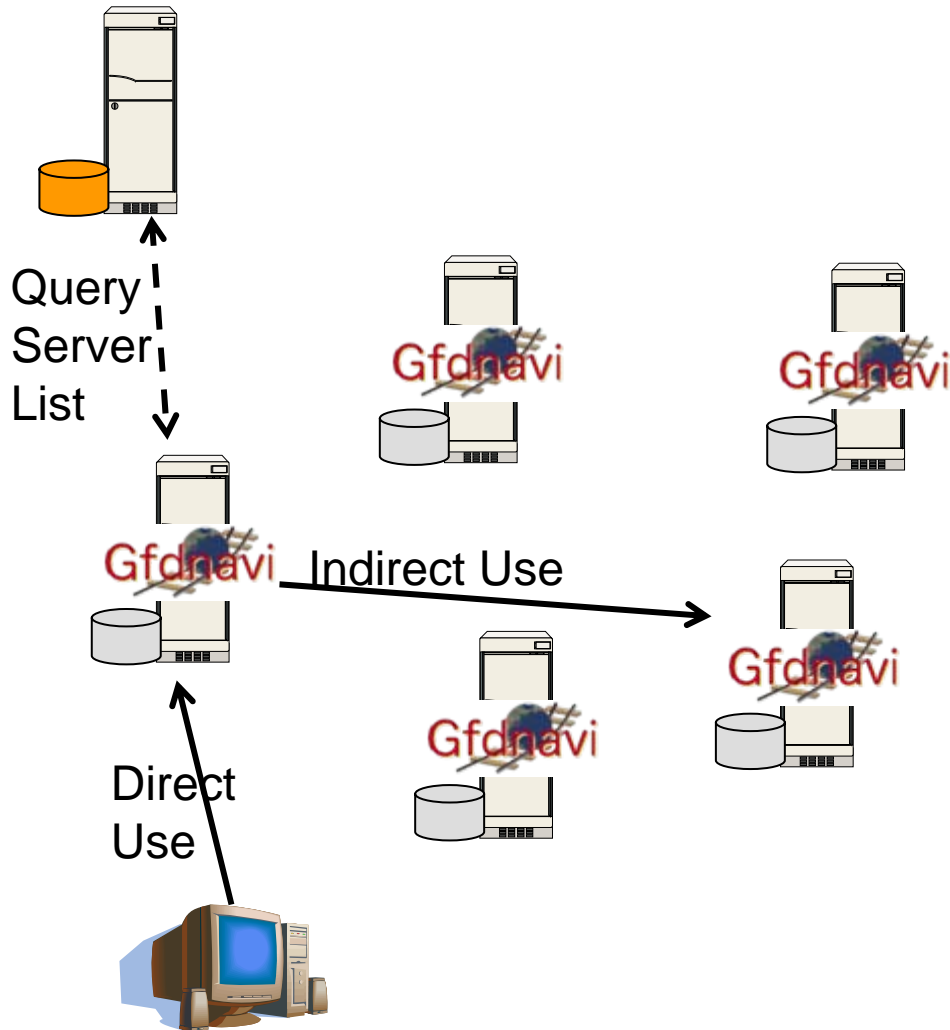


(b) Gfdnavi: one can register any peer by running a Gfdnavi

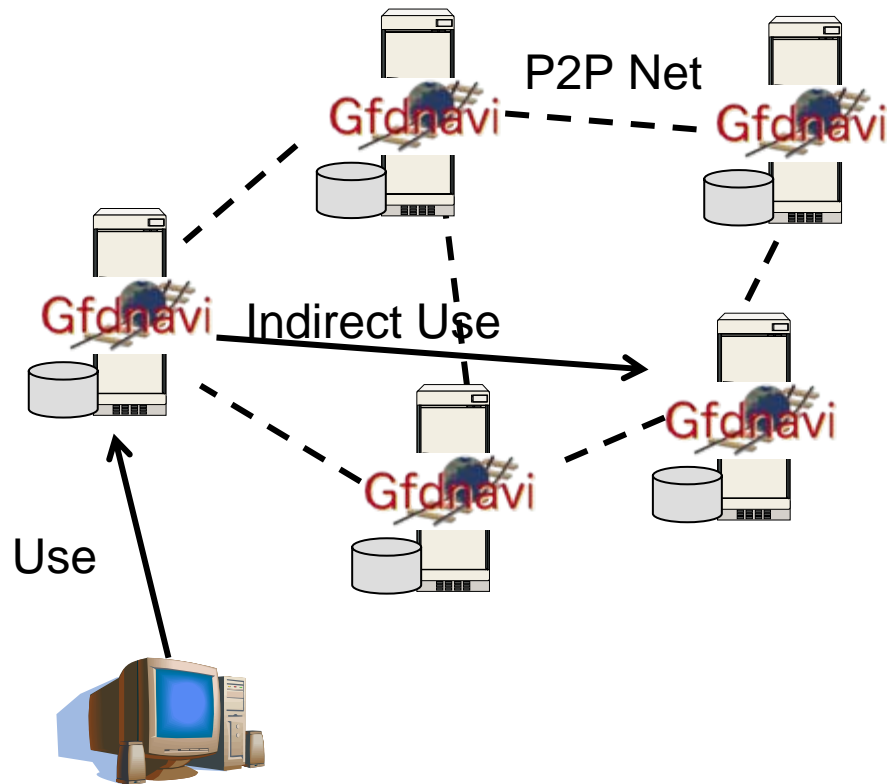


# P2P with directory server

Directory Server



# Overlay network by P2P



Currently tested by C. Watanabe by using  
Overlay Weaver (Java-based p2p library)  
and Rails' Action Web Service

- Decentralized p2p with distributed hash tables (DHT)



copy from old slides

last revised: 2004/06/08

# GPhys

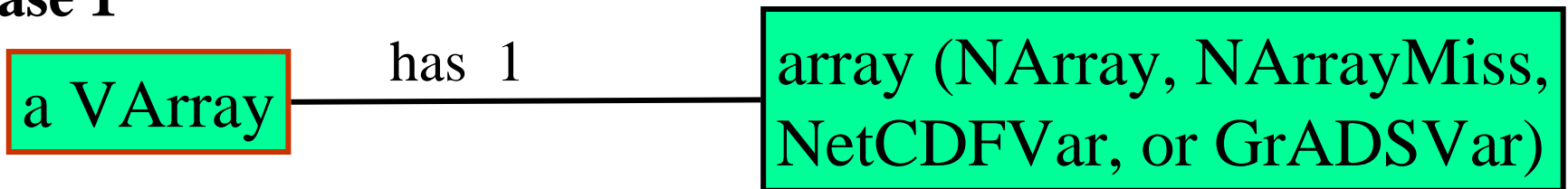
A class of gridded physical quantities

Takeshi Horinouchi (RISH, Kyoto  
Univ.)

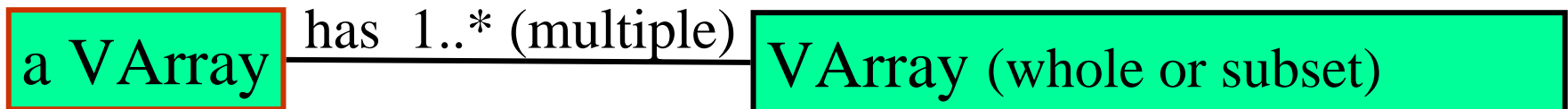
# VArray

- **Virtual Array**. A class of Ruby (written in pure ruby), which represents array data in GPhys
- A **VArray** object behaves as an array, but its contents can be on various media: (case 1) simply a multi dimensional array on memory (**NArray**), or data in a NetCDF file (in this case, a file pointer is stored), or GrADS data; (case 2) It can also represent a subset of another **VArray** or multiple **VArrays** tiled.
- Can have attributes as variables in NetCDF datasets
- In reality, NetCDF are handled by a subclass **VArrayNetCDF** etc, etc.

## Case 1

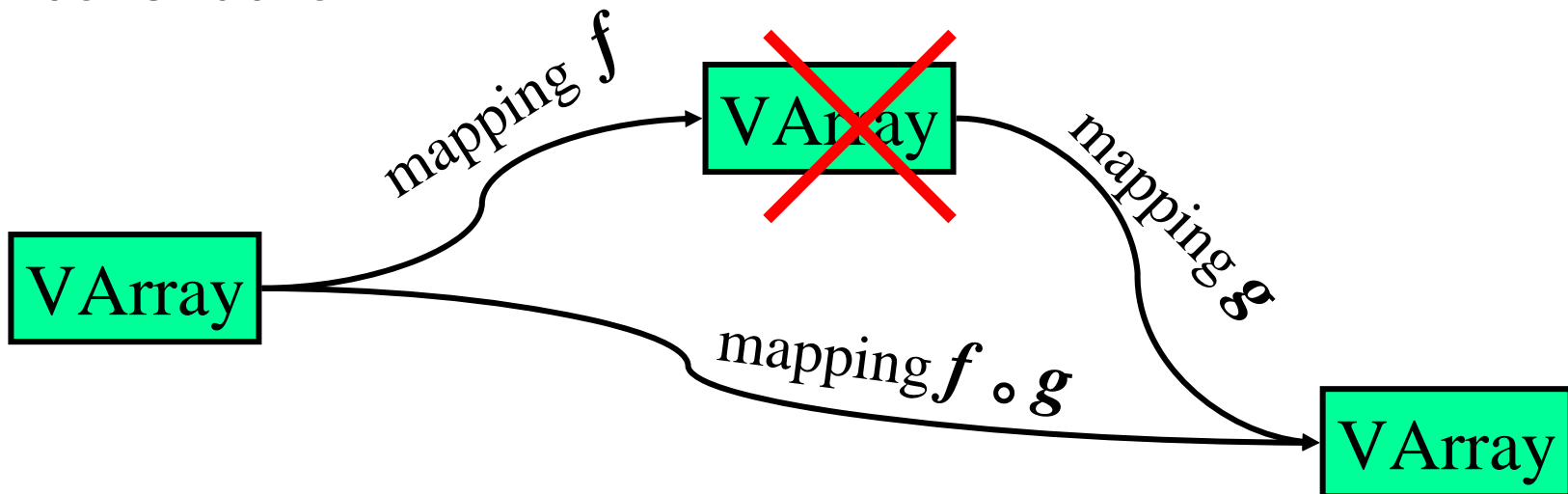


## Case 2



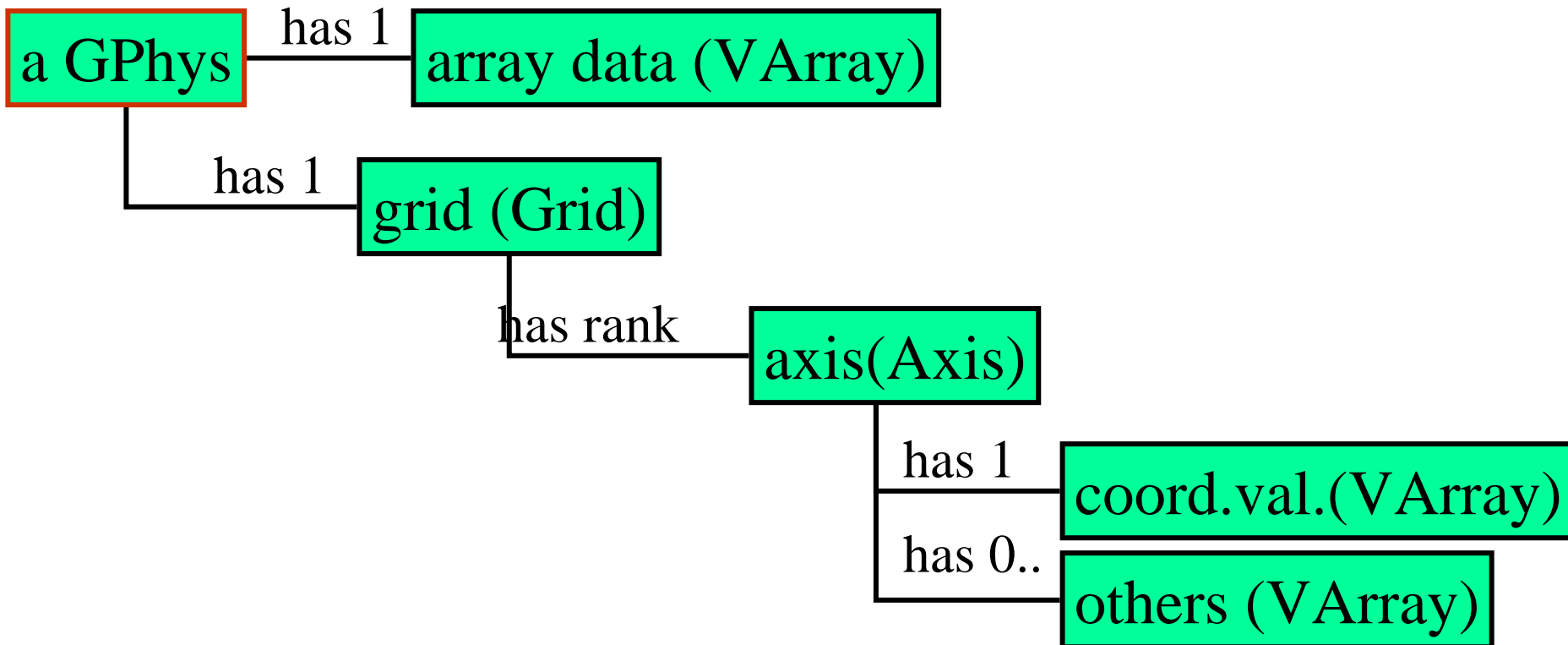
# subset mapping of VArray

- Always kept direct by compositing mappings, in order to prevent long chains (see the figure below).
- Subset slicing (by such as `va[0..10,3]`) is done by subset mapping, not by making actual data extraction, if not explicitly specified otherwise. Therefore,
  - Computationally efficient
  - Suitable for writing in subsets of data in files.
- In other words, actual data cutting is **deferred** until needed – to defer operations until needed is a policy of GPhys construction



# Structure of GPhys

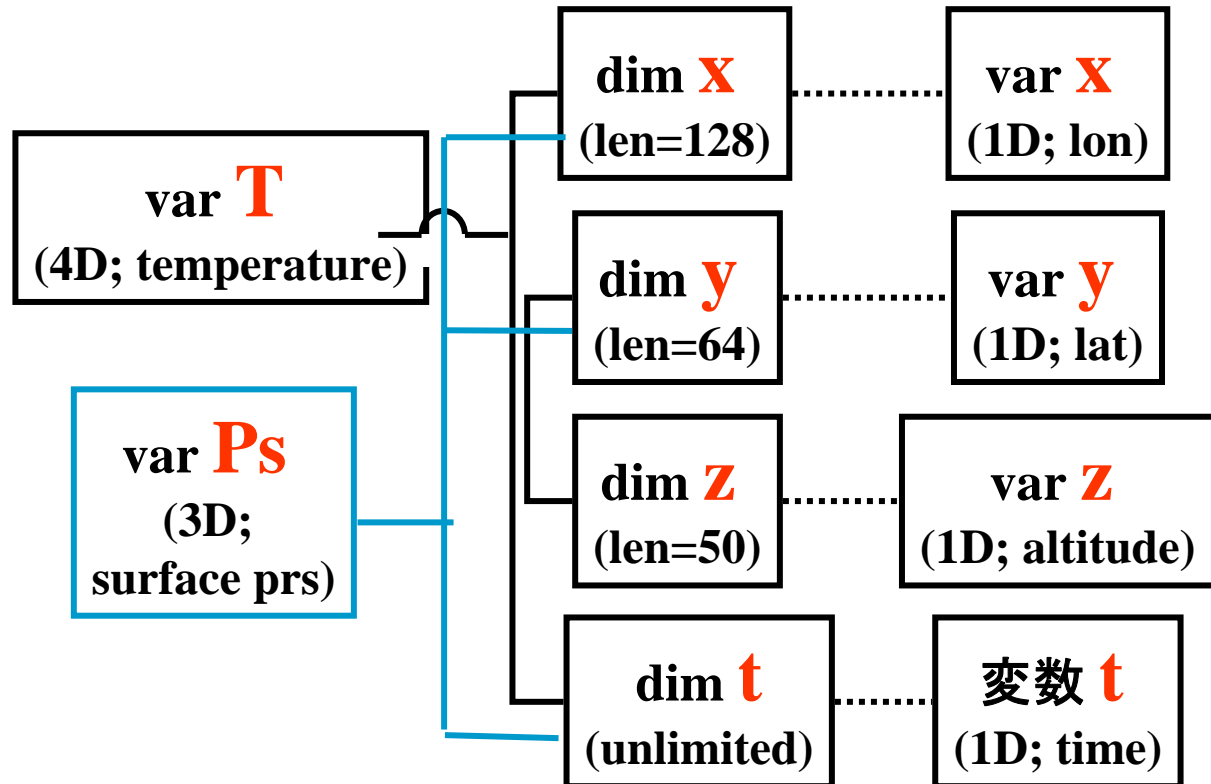
- Consists of a grid (coordinates) and multi dimensional array data
- Can conduct mathematical operations (a GPhys behaves like an numeric array)



# For your reference: Coordinates in NetCDF dataset

- Variables that have same names as dimensions hold coordinate values (locations)
- Weak point: this rule can be violated

**sample**



Can construct GPhys objects along the trees

# More on coordinates

- 3 cases are prepared
  - point sampling
  - cell type
  - simple sequence (though it's not physical)

point sampling — x — x — x — x — x —

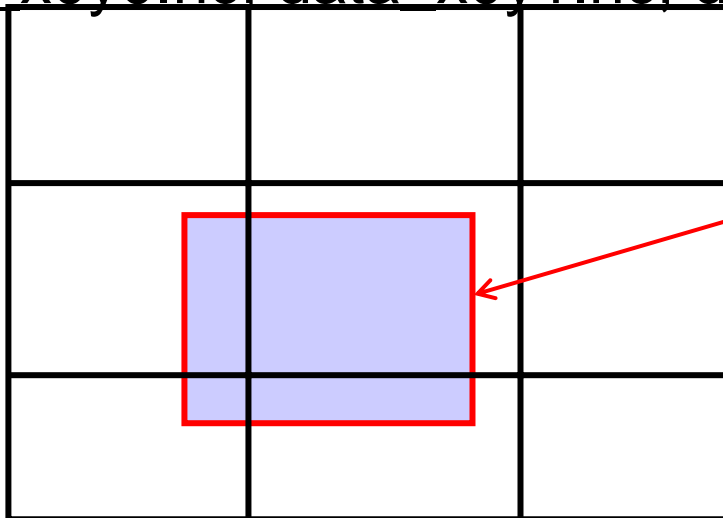
cell type | x | x | x | x | x |

(Here, coordinate variables can represent either boundaries (|) or representative locations such as centers (x).

- For instance, how to integrate along an axis is known by the axis. **GPhys** simply requests the integration to its **Grid**, and the **Grid** ask it to the corresponding **Axis**. By default, trapezoidal formula is used if point samples or cell boundaries. (can be changed by users)
- If NetCDF data are read, those types are configured if the convention used supports such discrimination. (so far, convention support is weak, though)

# Tiling

- Data divided into “tiles” can be treated as one consolidated GPhys object. Convenient to handle long time sequence divided by periods (such as by years) or outputs from parallel simulations on distributed-memory machines. Tiling is done by **VArrayComposite**.
- Subsets can be handled (see the figure below)
- May be applicable to parallel simulations in future?
- So far, automatic configuration is available only for NetCDF, by using an Array or Regexp (e.g., /data\_x(¥d)y(¥d).nc/ for data\_x0y0.nc, data\_x0y1.nc, data\_x1y0.nc, data\_x1y1.nc)



Subset specification  
of tiled data



# Big data handling

- Iterator to handle data too big to read on memory at once.
  - `GPhys::IO.each_along_dims_write` – the result also written in file (since the result of operations is often big too.)  
Another type of iterator is planned but yet to be implemented.
- Example:
  - Without the iterator:

```
in = GPhys::IO.open(infile, varname)
ofile = NetCDF.create(ofilename)
out = in.mean(0)      # now, the entire result is on memory
GPhys::IO.write( ofile, out )
ofile.close
```
  - With the iterator, taking the last dimension to make a loop:

```
in = GPhys::IO.open(infile, varname)
ofile = NetCDF.create(ofilename)
out = GPhys::IO.each_along_dims_write(in, ofile, -1){ |in_sub|
    [ in_sub.mean(0) ] # written in ofile each time
}
ofile.close
```

# Units of physical quantities

- Handled by NumRu::Units (by E Toyoda)
- mlt,div,etc.: handled as should be
- add,sub: the units of the first term is inherited
  - e.g., addition of [m] and [km] is done after multiplying the second term by 1000. Warning is made if the units are incompatible (in that case, no conversion is made).

- Introduced a scalar numeric class with units  
UNumeric

- GPhys, VArray, and UNumeric recognize one another (stronger to weaker in this order)
- Example: to multiply the Coriolis parameter with a GPhys object u representing winds [m/s]:

```
f = UNumeric[1e-4,"s-1"]
```

```
coriolis_frc = f * u    # then the units will be in m.s-2
```

# Distributed objects using dRuby

- Data service to remote clients
  - gphys-remote: a simple directory service (like the anonymous ftp, directories and data (in which GPhys objects can be defined) under a top directory is made accessible to remote hosts.
  - gave (GUI): can connect to gphys-remote server