Matlab GUIs for Ocean Satellite Data Products

Peter Cornillon

Graduate School of Oceanography/University of Rhode Island

OPeNDAP Developer's Meeting
21 February 2007

Acknowledgements

- Meri Sheremet - GSO/URI
- Christian Buckingham - GSO/URI

This work has been funded by the NASA/REASoN program, and by NOPP via the University of Miami.

Acknowledgements

- Meri Sheremet - GSO/URI
- Christian Buckingham - GSO/URI

This work has been funded by the NASA/REASoN program, and by NOPP via the University of Miami.

Acknowledgements

- Meri Sheremet - GSO/URI
- Christian Buckingham - GSO/URI

This work has been funded by the NASA/REASoN program, and by NOPP via the University of Miami.

Outline

Outline

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

Data Set Specific GUIs

URI, OPeNDAP, PMEL, UCAR, GSFC, JPL and RSS funded by NASA to develop

A Thematic Data Portal to Satellite-Derived Ocean Surface Properties

with particular attention on *Discovery and Access*

For this project, a suite of GUIs for satellite-derived ocean properties.

- Sea surface temperature
  - MODIS
  - AVHRR Pathfinder 4km
  - GOES
  - AIRS
- Surface winds and wind stress
  - NSCAT
  - QuikSCAT
- Ocean color
- Sea surface height
- Precipitation

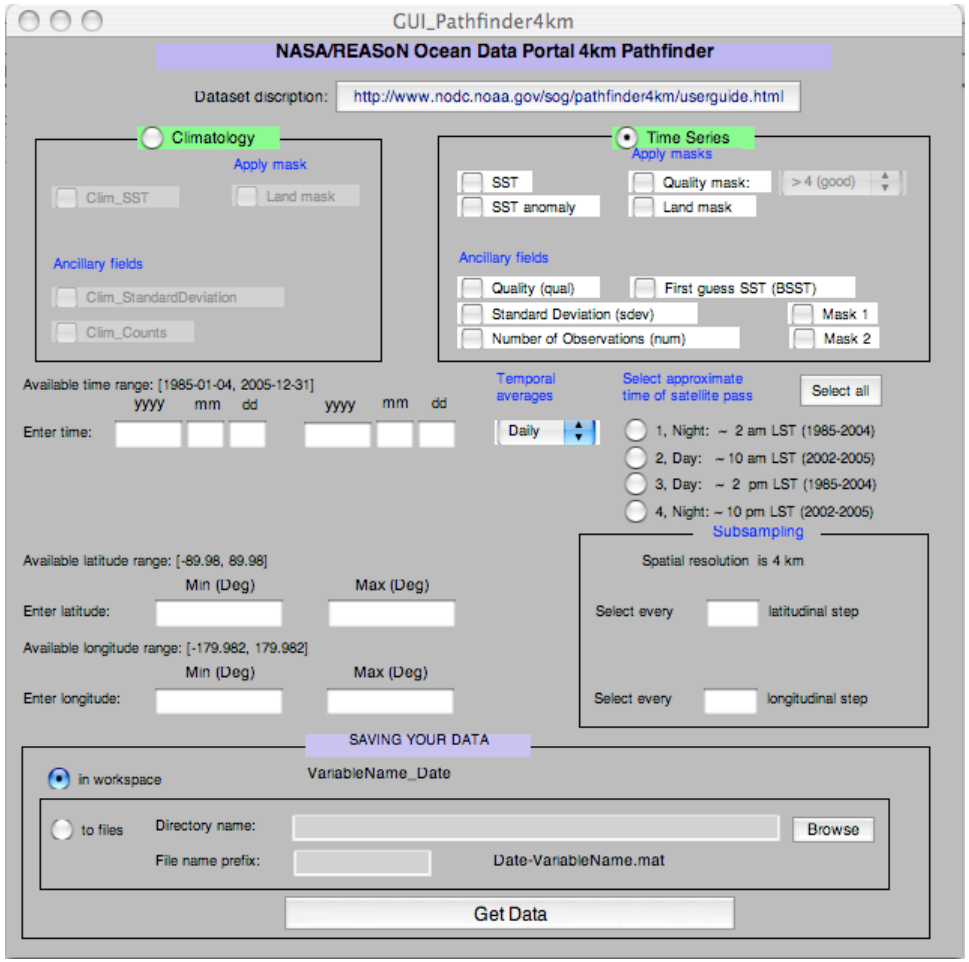
(A GUI also exists for HYCOM $\frac{1}{12}^{\circ}$ North Atlantic output.)

GUI Characteristics: Basics, Look and Feel

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.

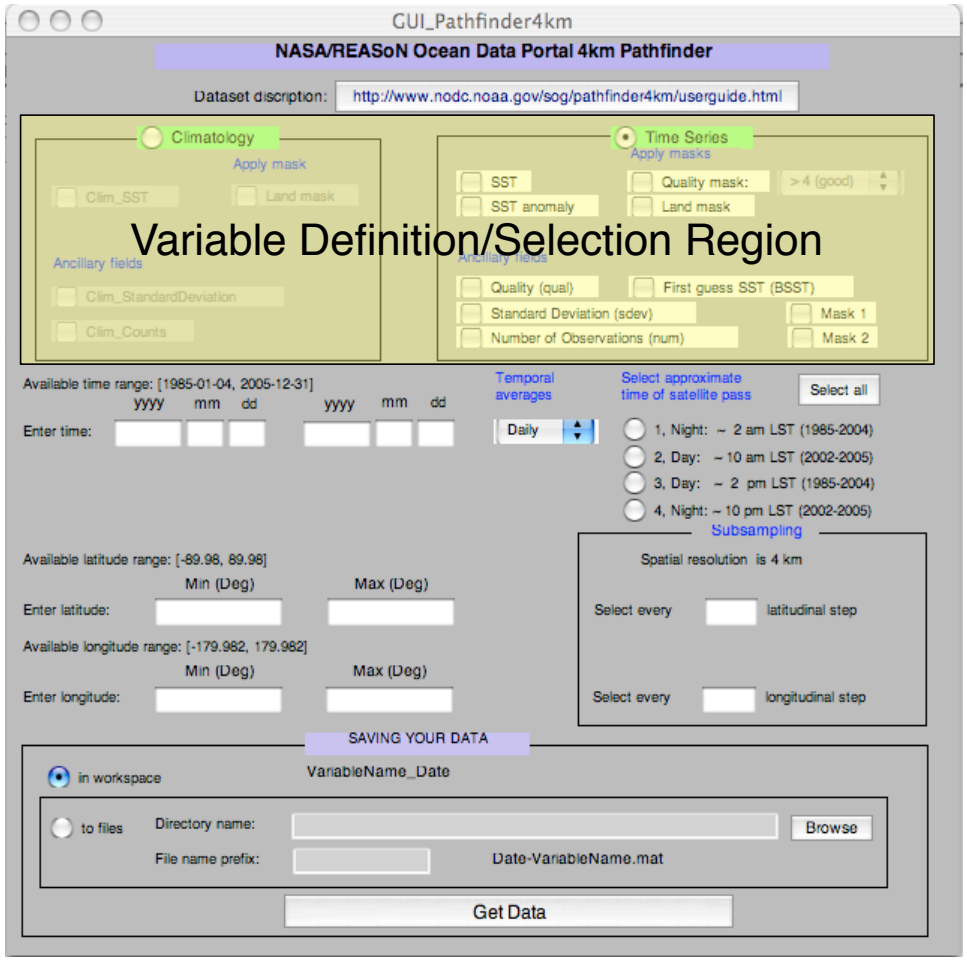
GUI Characteristics: Basics, Look and Feel

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.



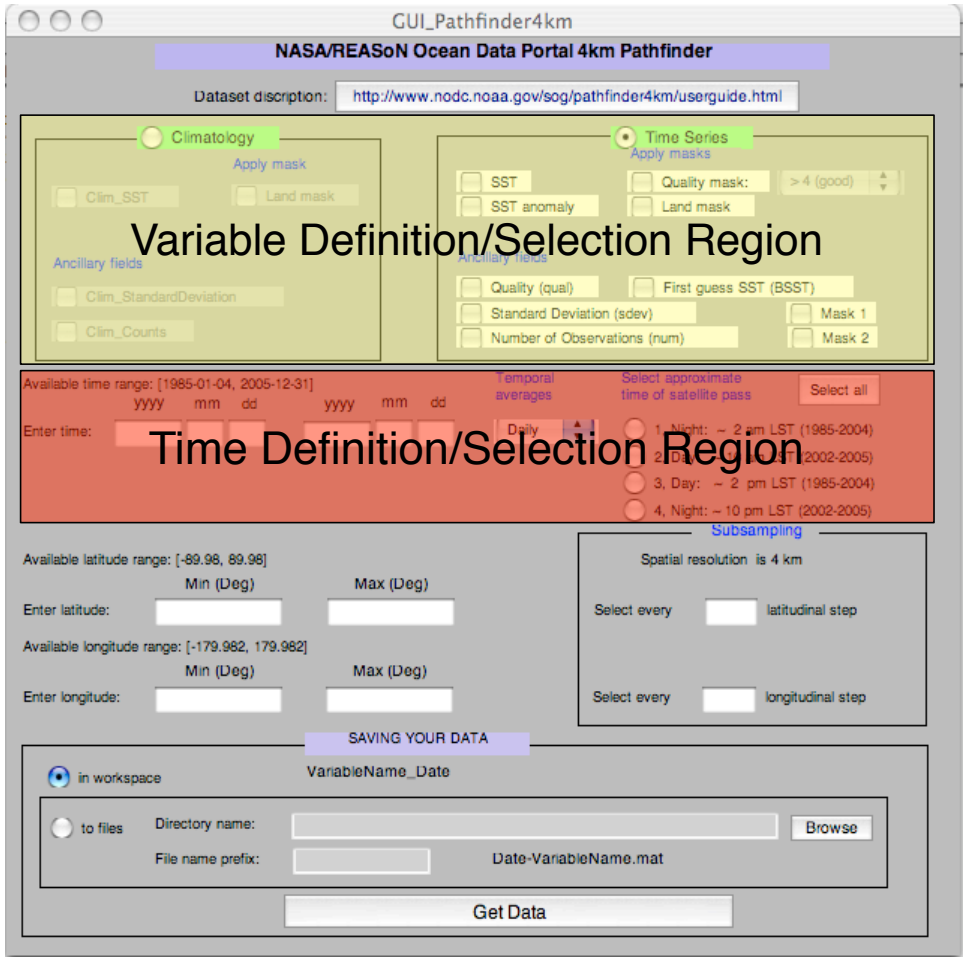
GUI Characteristics: Basics, Look and Feel

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.



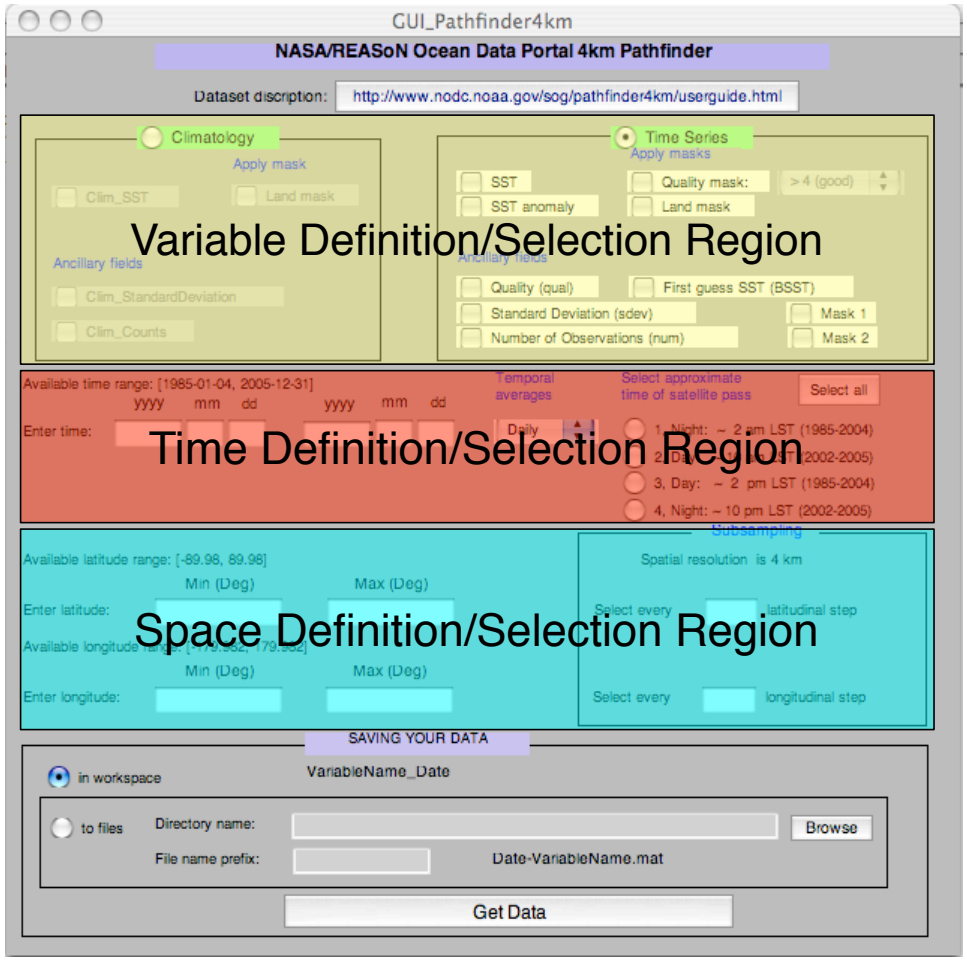
GUI Characteristics: Basics, Look and Feel

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.



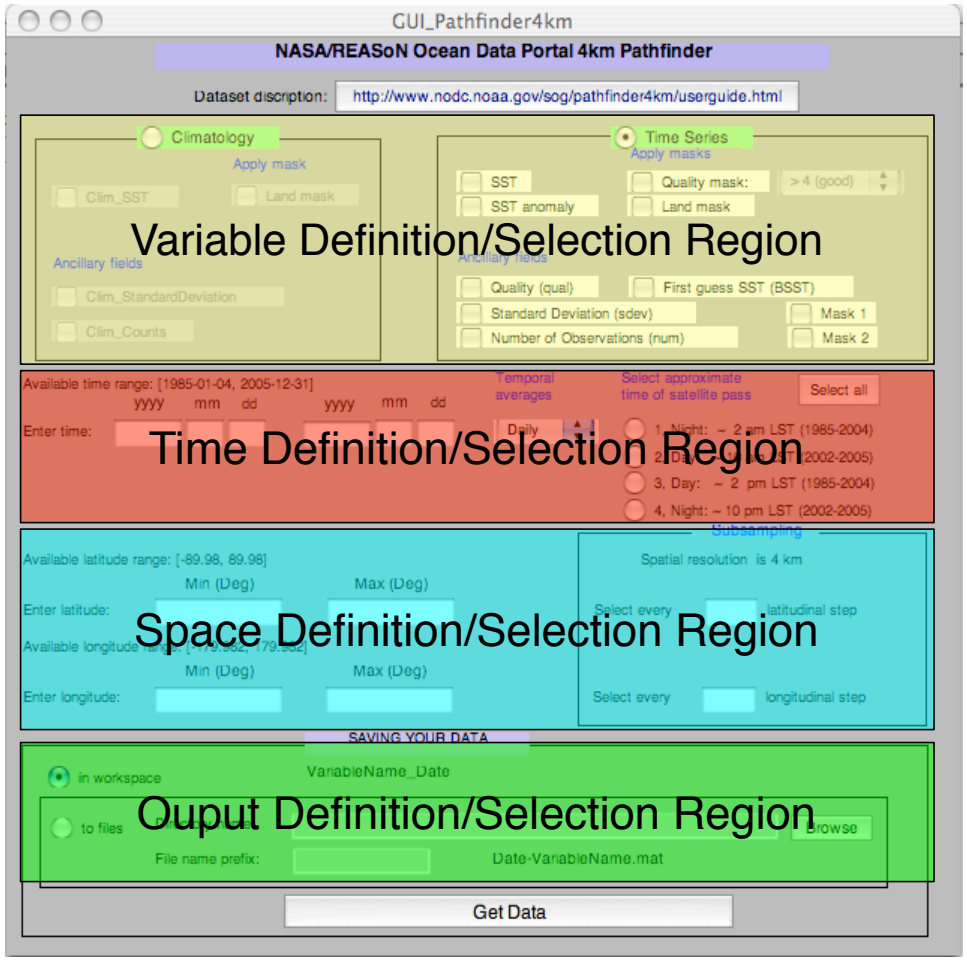
GUI Characteristics: Basics, Look and Feel

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.



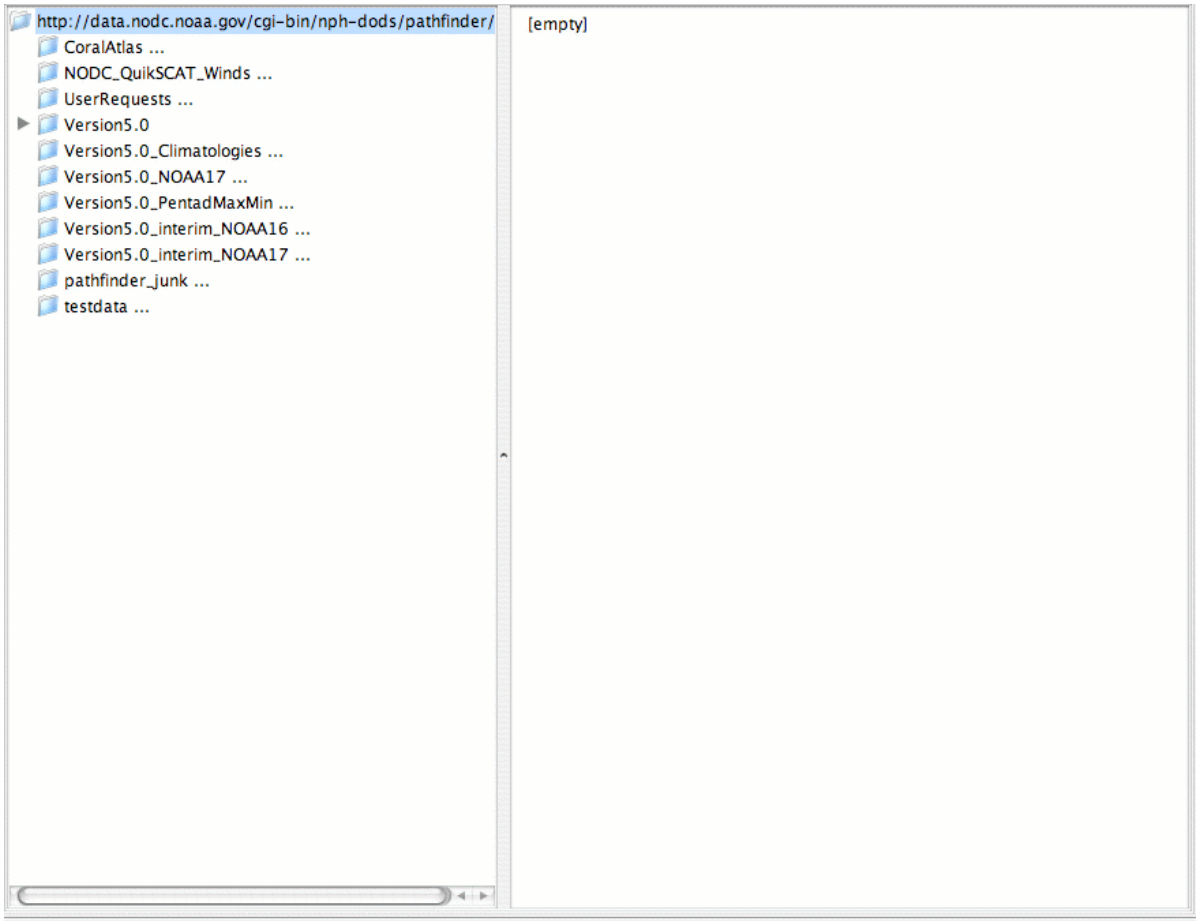
GUI Characteristics: Basics, Look and Feel

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.



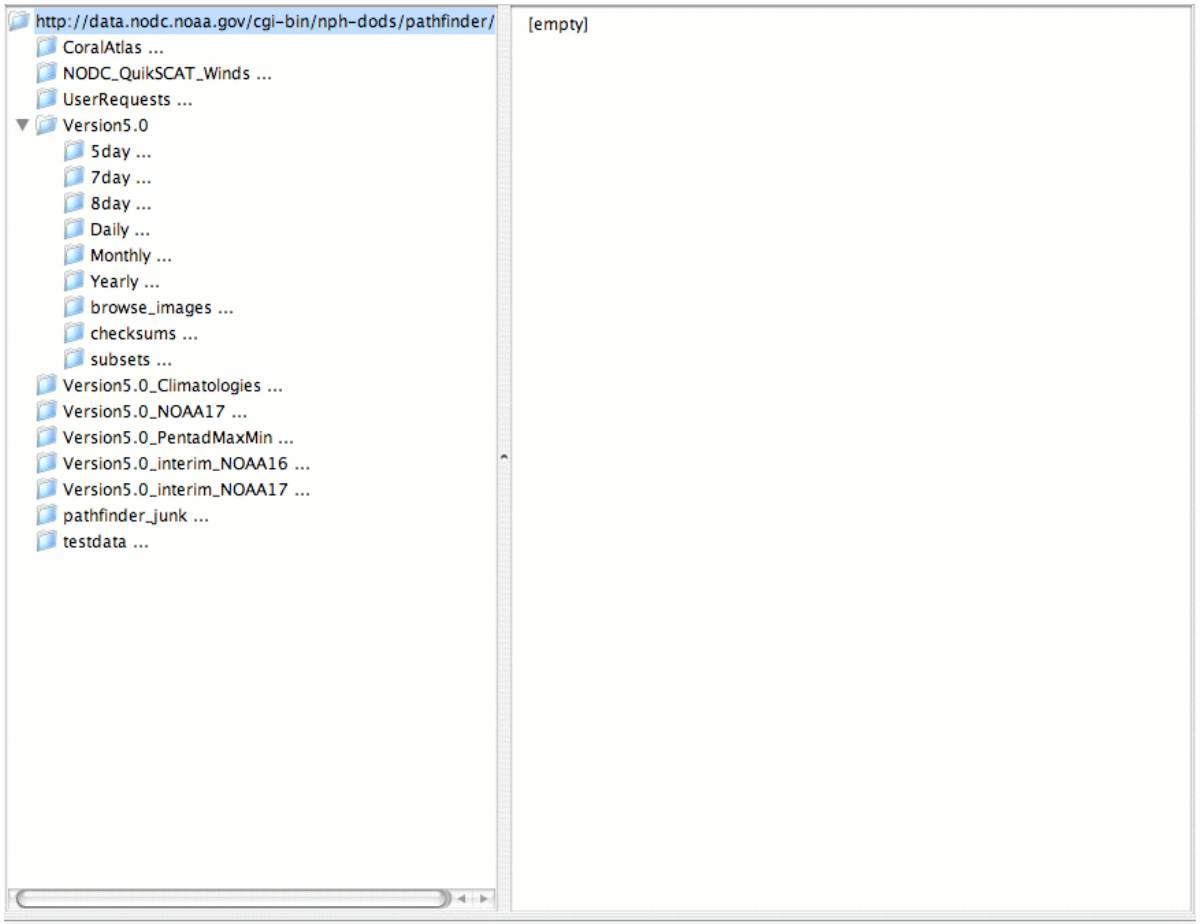
GUI Characteristics: Expose Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)



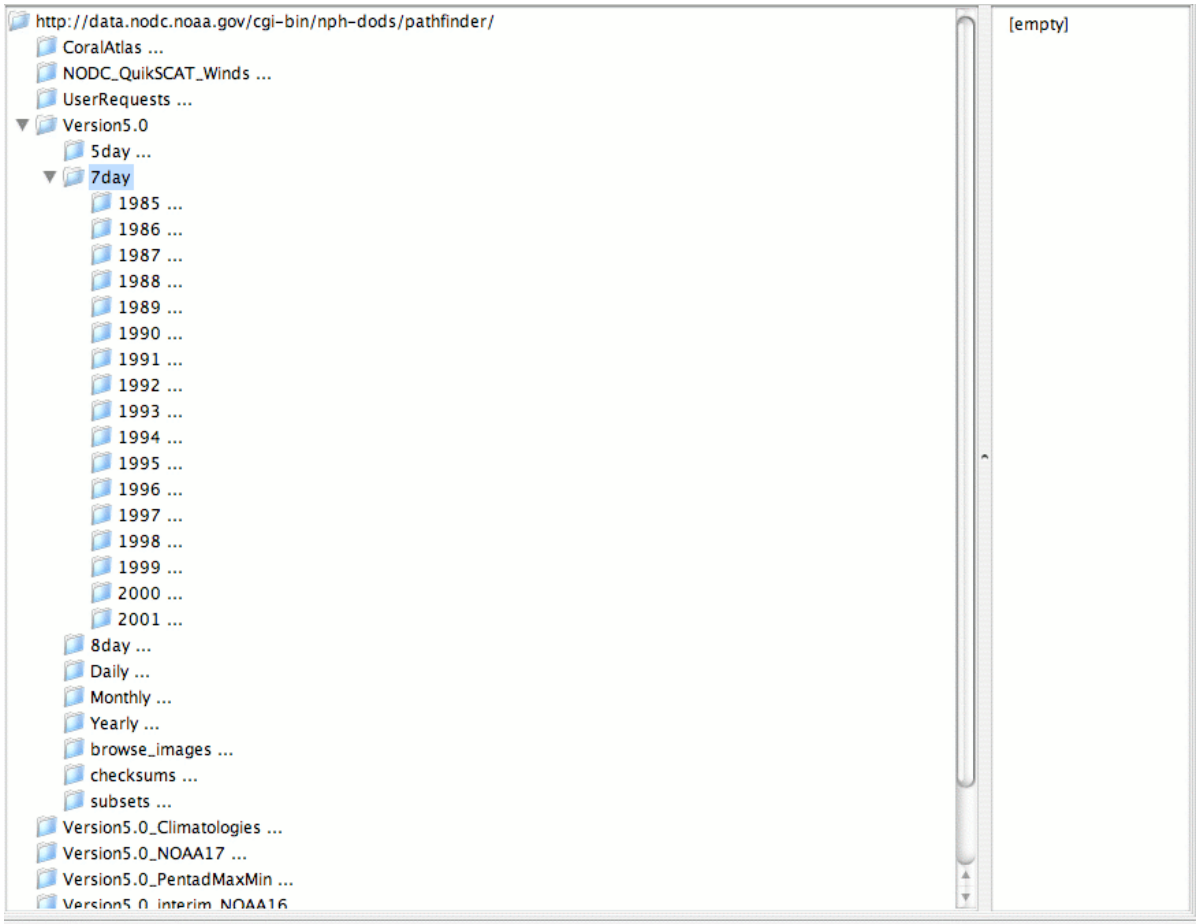
GUI Characteristics: Expose Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)



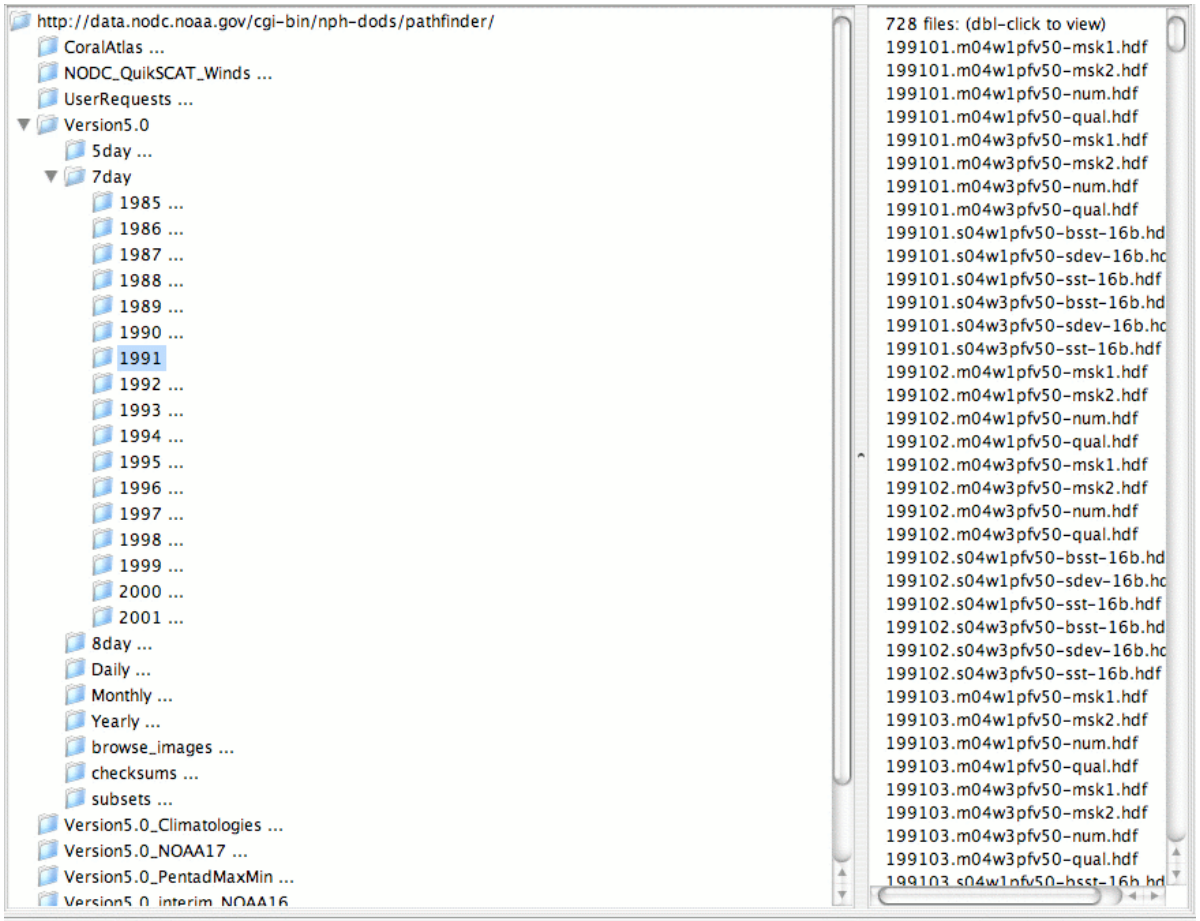
GUI Characteristics: Expose Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)



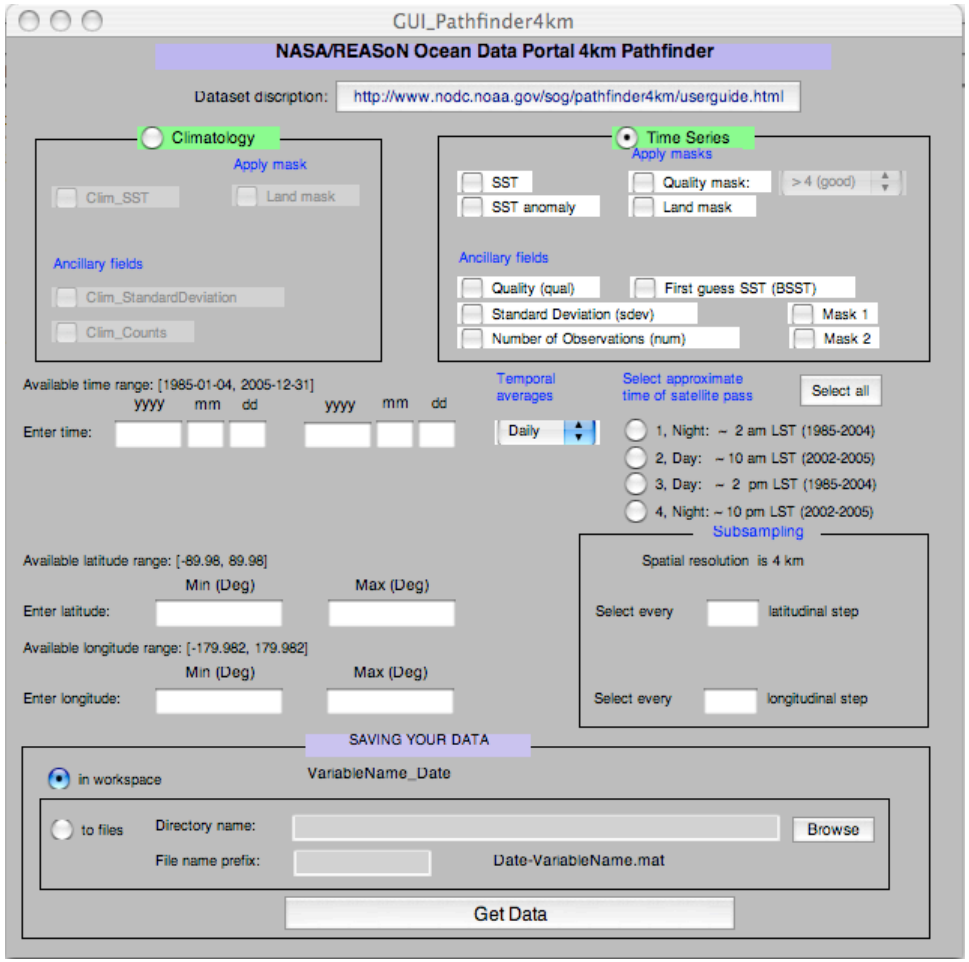
GUI Characteristics: Expose Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)



GUI Characteristics: Expose Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)



GUI Characteristics: Exposes Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)
- The underlying programs can be called as functions.
  - This means that the semantics used in the GUIs can be used in functions.
  - That the functions can be used in scripts.

GUI Characteristics: Exposes Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)
- The underlying programs can be called as functions.
  - This means that the semantics used in the GUIs can be used in functions.
  - That the functions can be used in scripts.

GUI Characteristics: Exposes Complicated Data Sets

- Written in Matlab, with plans to port to IDL.
- Provide for a similar look and feel across data sets.
- Expose the underlying structure of the archive. (What is a data set?)
- The underlying programs can be called as functions.
  - This means that the semantics used in the GUIs can be used in functions.
  - That the functions can be used in scripts.

Outline

Outline

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

Would Like Consistency in the Returned Data

- All of the GUIettes access multidimensional data:
  - 3D - **lon, lat, time**
    - MODIS, Pathfinder, scatterometer
  - 4D - **lon, lat, altitude (or depth), time**
    - AIRS, HYCOM
- The multidimensional nature of the data gave rise to two problems:
  - Structure name in the Matlab workspace
  - Structure of these structure

$$OPeNDAP\_0001. \begin{cases} temp - [1024x2096x12x2000] \\ temp\_error - [1024x2096x12x2000] \\ temp\_count - [1024x2096x12x2000] \\ Longitude - [1024] \\ Latitude - [2096] \\ Depth - [12] \\ Time - [2000] \\ URL \\ Metadata \end{cases}$$

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure Name Problem

Options

- **MODIS_yyyymmdd** – A name based on date.
  - Many different date/time representations – MODIS_20040110, MODIS_2004010, . . .
  - Cumbersome for multiple passes per day; requires date and time: MODIS_200401102315
  - Subsequent request can overwrite the structure, if the data/time are the same.
  - Possibility of lots of structures – a 4x4 pixel region for 1000 days $\Rightarrow$ 1000 structures in the user's workspace.
- **OPeNDAP_rrrr** –Some generic, non-descriptive name for the structure. (rrrr is the request number and is incremented for every read.)
  - Not user friendly.
- We are converting to the generic name representation plus a function to convert it to date/time
  - Provides for a user friendly response.
  - Avoids overwriting structures
  - Allows for different date/time naming conventions.
  - Puts the work in the function not in the GUIette.

The Structure of the Structure Problem

$$OPeNDAP\_0001. \begin{cases} temp - [1024x2096x12x2000] \\ temp\_error - [1024x2096x12x2000] \\ temp - [1024x2096x12x2000] \\ Longitude - [1024] \\ Latitude - [2096] \\ Depth - [12] \\ Time - [2000] \\ URL \\ Metadata \end{cases}$$

The Structure of the Structure Problem

$$OPeNDAP\_0001. \begin{cases} temp - [1024x2096x12x2000] \\ temp\_error - [1024x2096x12x2000] \\ temp - [1024x2096x12x2000] \\ Longitude - [1024] \\ Latitude - [2096] \\ Depth - [12] \\ Time - [2000] \\ URL \\ Metadata \end{cases}$$

$$nnnn = 1 : 2400 \qquad OPeNDAP\_nnnn. \begin{cases} temp - [1024x2096] \\ temp\_error - [1024x2096] \\ temp - [1024x2096] \\ Longitude - [1024] \\ Latitude - [2096] \\ Depth - 3 \\ Time - 127 \\ URL \\ Metadata \end{cases}$$

The Structure of the Structure Problem (continued)

We are leaning toward 2D fields, although another option is native

Outline

Motivation

- UCSB, URI and OPeNDAP have been funded by NSF to explore data discovery and access in a distributed environment.
  - Alexander Digital Library discover + OPeNDAP data access
- As part of this project, we are looking at:
  - The organization of geospatial data, and
  - The metadata found with OPeNDAP accessible data sets.

Motivation

- UCSB, URI and OPeNDAP have been funded by NSF to explore data discovery and access in a distributed environment.
  - Alexander Digital Library discover + OPeNDAP data access
- As part of this project, we are looking at:
  - The organization of geospatial data, and
  - The metadata found with OPeNDAP accessible data sets.

Motivation

- UCSB, URI and OPeNDAP have been funded by NSF to explore data discovery and access in a distributed environment.
  - Alexander Digital Library discover + OPeNDAP data access
- As part of this project, we are looking at:
  - The organization of geospatial data, and
  - The metadata found with OPeNDAP accessible data sets.

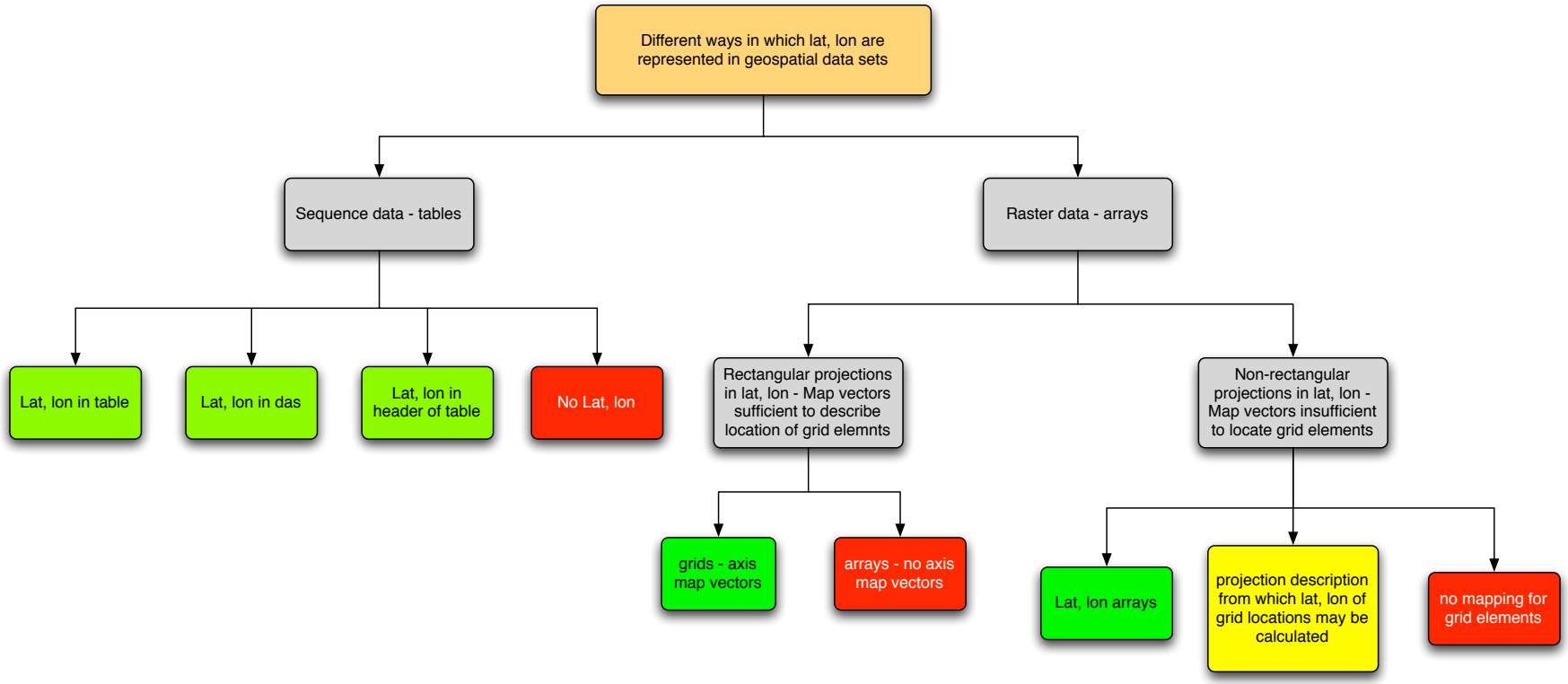
Motivation

- UCSB, URI and OPeNDAP have been funded by NSF to explore data discovery and access in a distributed environment.
  - Alexander Digital Library discover + OPeNDAP data access
- As part of this project, we are looking at:
  - The organization of geospatial data, and
  - The metadata found with OPeNDAP accessible data sets.

Motivation

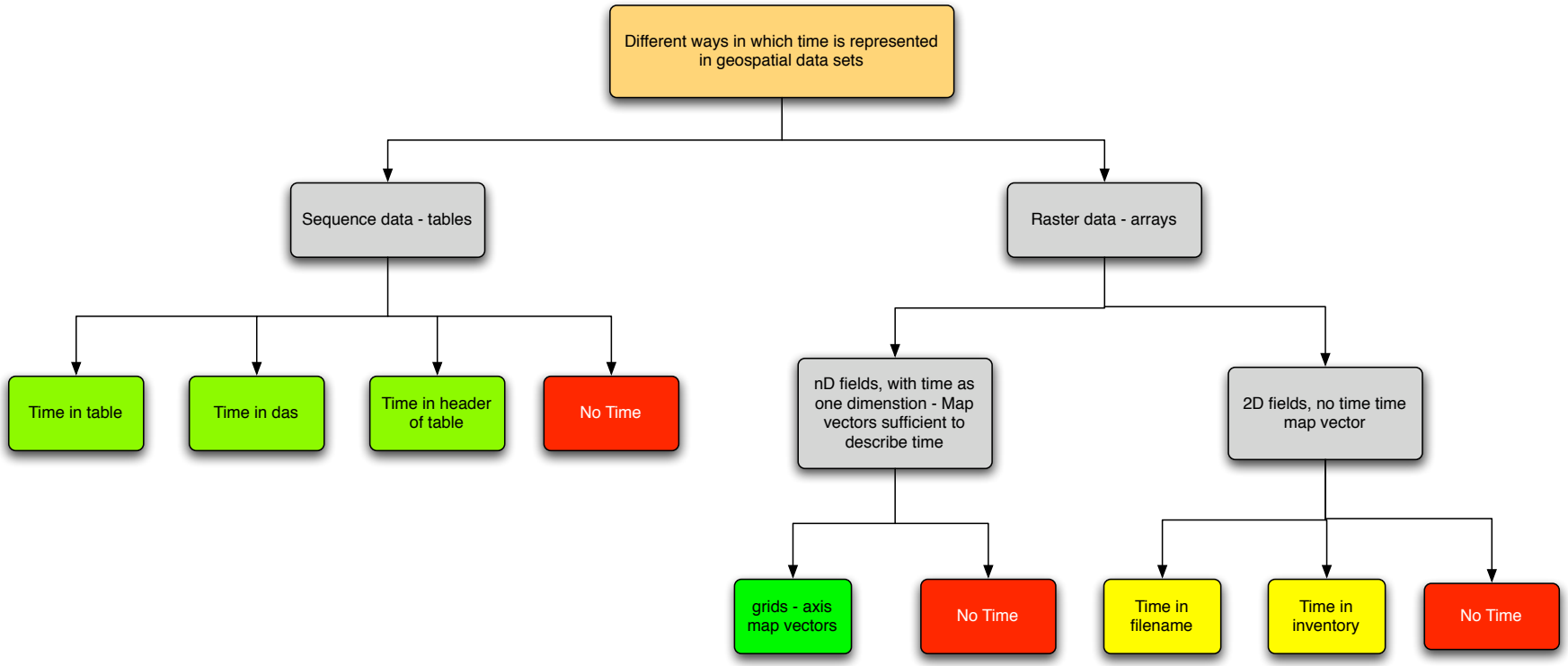
- UCSB, URI and OPeNDAP have been funded by NSF to explore data discovery and access in a distributed environment.
  - Alexander Digital Library discover + OPeNDAP data access
- As part of this project, we are looking at:
  - The organization of geospatial data, and
  - The metadata found with OPeNDAP accessible data sets.

Spatial Structure of Data Sets



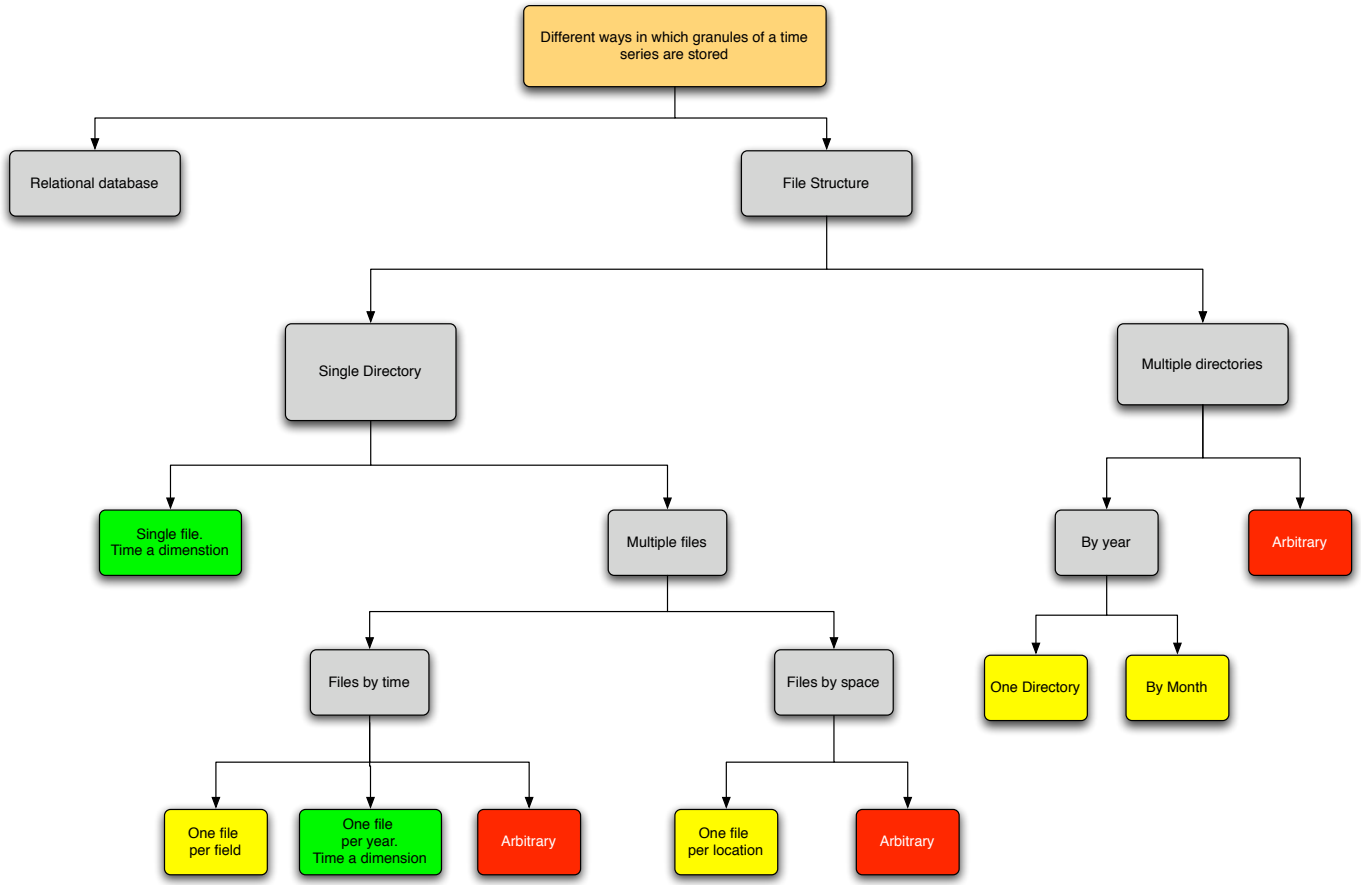
- Green rectangles mean lat, lon are readily available (assuming that one can associate them with the variable names used).
- Yellow rectangles mean that lat, lon are accessible, but require work.
- Red rectangles mean that lat, lon information is not accessible. Bad, bad, bad

Temporal Structure of Data Sets



- Green rectangles mean time is readily available (assuming that one can associate time with the variable name used).
- Yellow rectangles mean that time is accessible, but requires work.
- Red rectangles mean that time information is not accessible. Bad, bad, bad

Organizational Structure of Geospatial Data Sets



- Green rectangles mean the structure is very easy to parse.
- Yellow rectangles mean mean the structure is pretty simple to parse.
- Red rectangles mean that the structure is impossible to parse. Bad, bad, bad

Feedback

I'd appreciate any feedback on the structures (spatial, temporal or archive) that I have presented.