CCMC Data Standards
Conversion, Access, & Interpolation
Using Kameleon

Marlo Maddox
http://ccmc.gsfc.nasa.gov

NASA Goddard Space Flight Center
Models Covering the Entire Domain

- PFSS
- WSA PF+CS/WSA-IH
- MAS/CORHEL MAS
- Exospheric Solar Wind
- Heliospheric Tomography
- ENLIL
- SWMF

- SC/IH/BATS+RCM/GITM
- UPOS RB
- CMIT-LFM
- BATS
- Open GCCM
- Fok RC/RB
- Weimer2K
- SAMI2
- USU-GAIM
- OTIP
- AbbyNormal
Working With Unique Model Output

- No rules for standard model interfaces
- Each new model has unique output format
- Developer/user needs to become familiar with internal structure of each output file
- Custom read routines to access model data
- Data typically is not self descriptive
- Reduces portability and reuse of
  - Data output itself
  - Tools created to analyze data

Storage
Model Output Stored In Different Formats

```
model 1
data
model 1
tool 1
interface
model 1
tool 2
interface
model 1
tool m
interface
tool 1
tool 2
tool m

model 2
data
model 2
tool 1
interface
model 2
tool 2
interface
model 2
tool m
interface
tool 1
tool 2
tool m

model 3
data
model 3
tool 1
interface
model 3
tool 2
interface
model 3
tool m
interface
tool 1
tool 2
tool m

...

model n
data
model n
tool 1
interface
model n
tool 2
interface
model n
tool m
interface
tool 1
tool 2
tool m
```

\(n \times m\) interfaces required
Standardizing Unique Model Output

- Original output can be preserved
- Standard format for storage, coupling, visualization, & dissemination
- Model developers continue to have freedom of choice
- Ensures compatibility between models for coupling
- Ground work for which standard, reusable interfaces and tools can be developed

n + m interfaces required
Handling Unique Model Output With Kameleon

Original model output in unique storage formats is converted into a standard science data format with descriptive metadata elements added. This self-descriptive and platform-independent standardized model output is then accessed through a user-friendly software library, providing a single interface to all standardized data.
Kameleon Supported Models At The CCMC

Standardized Model Output
- Available
- Beta Version
- Under Development

- PFSS
- WSA PF+CS/WSA-IH
- Exospheric Solar Wind
- WSA-PF+CS/WSA-IH
- Heliospheric Tomography
- MAS/CORHEL MAS
- ENLIL
- SWMF

- SC/IH/BATSRUS+RCM/GITM
- CMIT-LFM
- BATSRS
- Open GCCM
- Fok RC/RB
- Weimer2K
- SAMI2
- USU-GAIM
- CTIP
- Abby Normal
Kameleon Software Suite Overview

**Kameleon Software Suite**

- Converts and stores disparate data sets into self-descriptive standardized files
- Comprehensive metadata model applied to each file
- Library provides direct data access to converted space weather data
- Interpolation, metadata extraction, & derived variable calculations
- Library callable from any C-supported programming language or application
- Promotes data reuse & code reuse
- Can support/be applied to more than model output - Magnetogram Synthesis (P. Macneice)
**Kameleon Converter**

- Ingests supported data files and converts original data into a specific scientific data standard format
  - Platform independent
- Adds descriptive meta elements to each converted data files
  - Grid description Information
  - Coordinate System descriptions
  - Detailed variable descriptors
  - General and Model specific descriptive information

**Kameleon Library**

- Provides access and interpolation functionality to Kameleon converted data files
  - Standard interface to Multiple and diverse data sets
  - Masks complexity of underlying storage container
  - Efficient direct data access
- Spatial & temporal interpolation
- Query global & variable metadata attributes
- Several interfaces Provided
  - C, C++, FORTRAN, IDL
- Can be used in any C supporting application
Aside from the one-to-one data conversion, what additional metadata do we want to provide?

- Global
  - General description of the model / data
  - Coordinate system(s)
  - Grid Description
    - # of grids
    - # of dimensions
    - dimension size(s)
  - Date & Time Information

- Variable metadata - descriptive elements for each variable
  - Units
  - Actual & Valid Min/Max values
  - Masks Values

- Model Specific Metadata
- SPASE - Space Physics Archive Search and Extract Data Model
  - Computational Model Group
- UMICH - SWMF / Batsrus Data Standardization
# Kameleon Global Attributes

- README
- README_visualization
- model_name
- model_type
- generation_date
- original_output_file_name
- run_registration_number
- generated_by
- terms_of_usage
- grid_system_count

# Kameleon Variable Attributes

- valid_min
- valid_max
- units
- grid_system
- mask
- description
- is_vector_component
- position_grid_system
- data_grid_system
- actual_min
- actual_max

# Model Specific Attributes

- Additional grid descriptors
- Original output data or descriptors that don’t map to predefined attributes
- Any additional elements that are specific to a particular model or space weather domain
Kameleon Conversion Software Components

- **main read driver**
  - read model a routine
  - read model b routine
  - read model n routine

- **main write driver**
  - write model_to_structure
  - write structure to cdf
  - write structure to hdf5

- **main conversion routine**

- **model specific attribute list (.h)**
- **generic attribute list (.h)**
- **generic/default variable attribute list (.h)**
- **model specific attribute list (.h)**
- **model specific variable attribute list (.h)**

- **structure manager**
  - structure definitions (.h)

- **variable attributes**
- **variable names**

- **global/file attributes**

- **Model Variable List**

- **Registered Variables List**
  - CCMC_name
  - x, x_pos, xp
  - y, y_pos, yp

- **Model Data Assembled Into Standard Data Structures**

- **standard data files with common attributes and variable names for each registered model**
Standardized Attribute & Variable Structure Lists

Attribute List
- attribute 1
- attribute 2
- attribute 3
- ...
- attribute n

Attribute Structure
- attribute name
- attribute type
- attribute data type
- attribute value

Variable List
- variable 1
- variable 2
- variable 3
- ...
- variable n

Variable Structure
- variable name
- variable data type
- variable size
- data classification
- variable values
- valid min
- valid max
- units
- grid system
- mask
- description
- is vector component
- position grid system
- data grid system
- actual min
- actual max

Model Data Assembled Into Standard Data Structures

write from structure to standard format module
Populating the Structures

- Library of C routines that are used to populate the standard attribute and variable structures.

- Model Read/Write Routine

  - CCMC Structure Manager Library
    - init_ccmc_attribute_structure
    - init_ccmc_variable_structure
    - put_ccmc_attribute
    - put_ccmc_variable
    - update_ccmc_attribute_value
    - update_ccmc_variable_value
    - update_ccmc_variable_attribute_value
    - free_ccmc_attribute_structure
    - free_ccmc_variable_structure

- Write Model to Structure Routine

- Model Data Assembled Into Standard Data Structures

- write structure to standard format

  - standard data files with common attributes and variable names for each registered model
KAMELEON Access/Interpolation Library

KAMELEON Standardized Model Data

KAMELEON Access/Interpolation Library

Your Code/Application

KAMELEON Access & Interpolation Library

- open_cdf(cdf_name, 0);
- get_units( variable_name);
- interpolate_batsrus_cdf( variable1, X, Y, Z, 0, 0);
- interpolate_ucla_ggcm_cdf( variable1, X, Y, Z, 0, 0);
- interpolate_ctip_cdf( variable1, X, Y, Z, 0, 0);
- interpolate_enlil_cdf( variable1, X, Y, Z, 0, 0);
- close_cdf();
- gattribute_float_get( attribute_name );
- gattribute_char_get( attribute_name );
- init_time( data_path, start_time, end_time );
- time_interpolate( variable_name, time, X, Y, Z );
- vattribute_get( variable_name, attribute_name );

CDF Library

FORTRAN INTERFACE

Call from any C supported Programming Language:

- Fortran
- C/C++
- IDL
- OpenDx

- Java
- Perl
- Vtk
- Your App

Current Standardized Model Output Availability

- BATSRUS
- OpenGGCM / UCLA-GGCM
- CTIP
- ENLIL
- MAS (Beta Version)

Currently Supported Science Data Formats

- CDF 2.7
- CDF 3.0
- CDF 3.1
- HDF5 (under consideration)
KAMELEON Access/Interpolation Library

KAMELEON Standardized Model Data

KAMELEON Access/Interpolation Library

Your Code/Application

Call from any C supported Programming Language:
- Fortran
- C/C++
- IDL
- OpenDx
- Java
- Perl
- Vtk
- Your App

Current Standardized Model Output Availability
- BATSRUS
- OpenGGCM / UCLA-GGCM
- CTIP
- ENLIL
- MAS (Beta Version)

Currently Supported Science Data Formats
- CDF 2.7
- CDF 3.0
- CDF 3.1
- HDF5 (under consideration)
General Usage and Benefits

- Self descriptive data files
- Platform independent
- Promotes data sharing
- Speed and efficiency of direct data access
- Same interface regardless of model/data input
- Facilitates code reuse
- Kameleon library allows model data to be more easily integrated into existing analysis and software applications

… addresses some of the needs and requirements of “power-users”, as identified from the user feedback sessions from Monday.
Specific Usage and Benefits

- **CCMC:**
  - Runs-On-Request: Converted Data & Kameleon Download - to be automated
  - CCMC Visualization: Space Weather Explorer
  - CCMC Visualization: Space Weather View
  - Particle Tracing
  - Custom/Specialized Analysis Software: Field/Flow Line Tracing
  - Derived Library Add-On for Kameleon - D. Berrios

- **External Research & Analysis Packages:**
  - MAGIC - MAGnetogram Interpolation & Composition - Magnetogram Synthesis
  - Themis Support
  - Visbard Integration
  - Possible integration with CISMDx Viz tool

- Comparing Model data and observational data
- General data analysis - diverse set of users that have requested and used Kameleon
- Access/Interpolation library is highly configurable and expandable.
TODO List / Things To Remember

- Variable naming conventions
- Unit conversions
- Grid description refinement
- Coordinate transformations between native and target grid(s)
- Opening Multiple files in memory with targeted interpolation on specific data set
- Fulfilling expanded feature requests
- Extracting Kamelon converter structure manager for external use
- Refining access/interpolation library as feedback is acquired
- Identifying an extensive list of desired/requested routines and functionality
- Working with external groups - identifying methods to formally provide standardized model data along with the Kameeon access/interpolation library
- Configuration Management: Model, Converter, Access Library, Container versions
Summary

• Metadata is a key component.
  – clearly defined set of core metadata elements that are currently being implemented on Kameleon converted data sets
  – Recently started collaborating with SPASE Working Group
• Structure oriented architecture of Kameleon Converter ensures flexibility and expandability
  – Internal kameleon conversion functionality can ultimately be used by external developers
• Kameleon Software Suite currently supports:
  – BATSRUS, OpenGGCM, CTIP, ENLIL, & MAS
  – Select Observational data sets for MAGIC / Magnetogram Synthesis
• Kameleon access/interpolation library key features:
  – Interface to easily extract global & variable metadata
  – Time interpolation for MHD data sets
  – Fortran interface
  – IDL interface
  – Derived Library
CCMC Data Standards
Data Conversion, Access, & Interpolation
Using Kameleon

Marlo Maddox

http://ccmc.gsfc.nasa.gov

NASA Goddard Space Flight Center
Space Weather Models

Solar Interior (SI)

Solar Atmosphere (SA)

Solar Wind (SW)

Magnetosphere (MG)

Plasmasphere (PL)

Ionosphere (IO)

Neutral Atmosphere (NA)

Inner Magnetosphere (IM)

Ring Current/Radiation Belt

Ionosphere Electrodynamics (IE)

patch-panel architecture
Data Format Standard Options

- CDF
- HDF, HDF4, HDF5
- NetCDF
- FITS
- GRIB
- BUFR
- GRADS
- Office Note 29
- Office Note 84
- VICAR
- PDS
- Open Dx Data Model
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main( int argc, char *argv[] )
{
    extern long init_time(char *, double *, double *);
    extern float time_interpolate(char*,double,float,float,float);
    long status;
    char data_path[750];
    char variable[10];
    float X, Y, Z;
    double time, start_time, end_time;
    float sample_time_interval;
    float time_interpolated_value;
    
    strcpy( data_path, argv[1] );
    strcpy( variable, argv[2] );
    X = atof( argv[3] );
    Y = atof( argv[4] );
    Z = atof( argv[5] );
    sample_time_interval = atof( argv[6] );

    status = init_time( data_path, &start_time, &end_time );
    
    printf("Simulation start_time:\t%f msec\n", start_time);
    printf("Simulation end_time:\t%f msec\n", end_time);

    for(time=start_time;time<=end_time;time+=sample_time_interval)
    {
        time_interpolated_value=time_interpolate(variable,time,X,Y,Z);
        printf( "%s [ %f, %f, %f ] @ %f milliseconds\t%f\n",variable,X,Y,Z,time,time_interpolated_value );
    }
    return 1;
}
```

C example of 4D time interpolation of CCMC standardized data using access/interpolation library
program f2c_interp_open_ggcm

c Three functions used to interpolate
c data from a specified batsrus cdf file
external f2c_open_cdf, f2c_close_cdf, f2c_interp_bats_cdf
c Variables to be used for interpolation and data extraction
character*150 cdf_file_path
real*8 x,y,z
real*8 interpolated_value
integer status
character*50 var_to_read

c --- set your actual path name here ---
cdf_file_path='open_ggcm.cdf '

c Open the cdf file
status=f2c_open_cdf(cdf_file_path)

c --- set your position values in GSE ---
x=-55.0
y=12.0
z=20.0

c --- set name of variable of interest ---
var_to_read='bx '

c --- call the interpolation routine ---
status=1f2c_interp_open_ggcm_cdf(x,y,z,interpolated_value,var_to_read)

c --- close the currently open cdf file
status=f2c_close_cdf(0)
write(*,*) var_to_read, interpolated_value
end