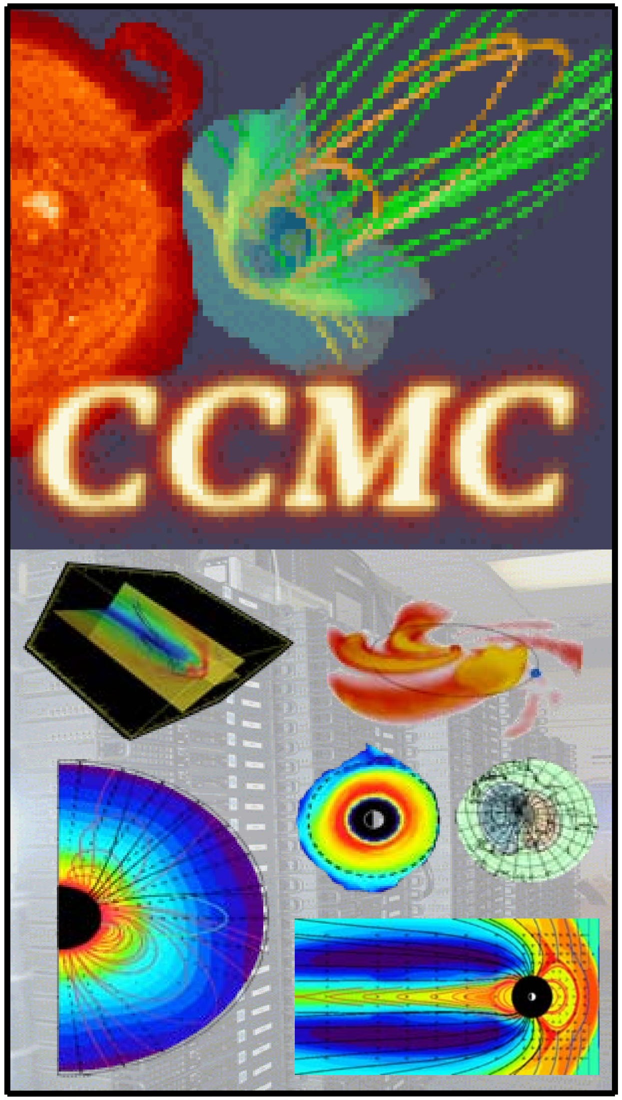


Data Format Standardization of Space Weather Model Output at The Community Coordinated Modeling Center

M. Maddox

NASA Goddard Space Flight Center

http://ccmc.gsfc.nasa.gov

Hilton
Clearwater Beach Resort

CCMC Workshop

October 11-14, 2005

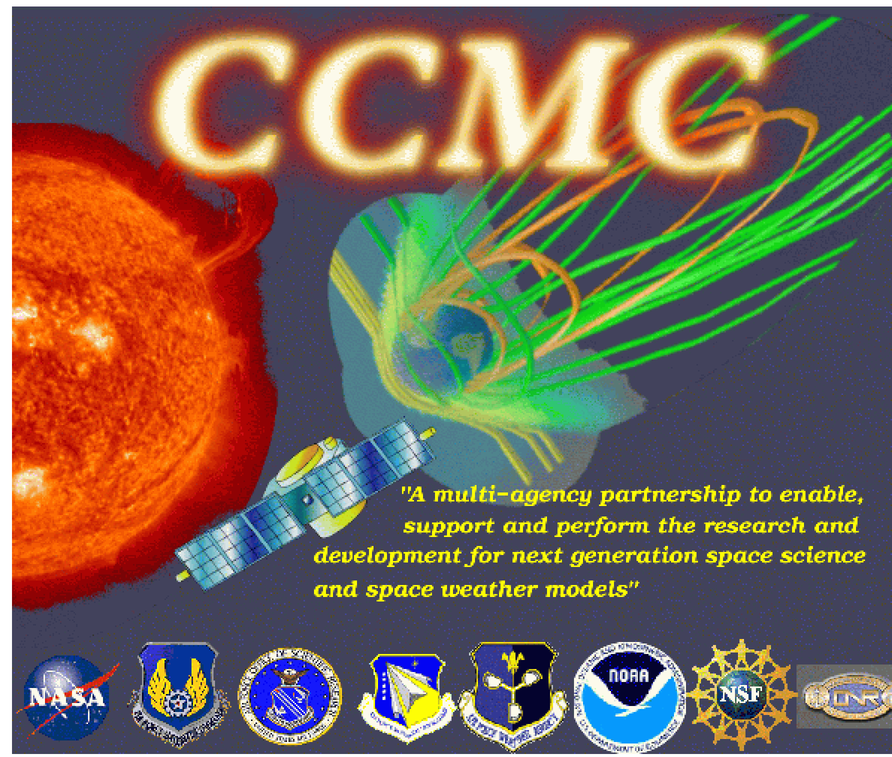
Tuesday, October 11th PM Session:

Frameworks and Data Infrastructure

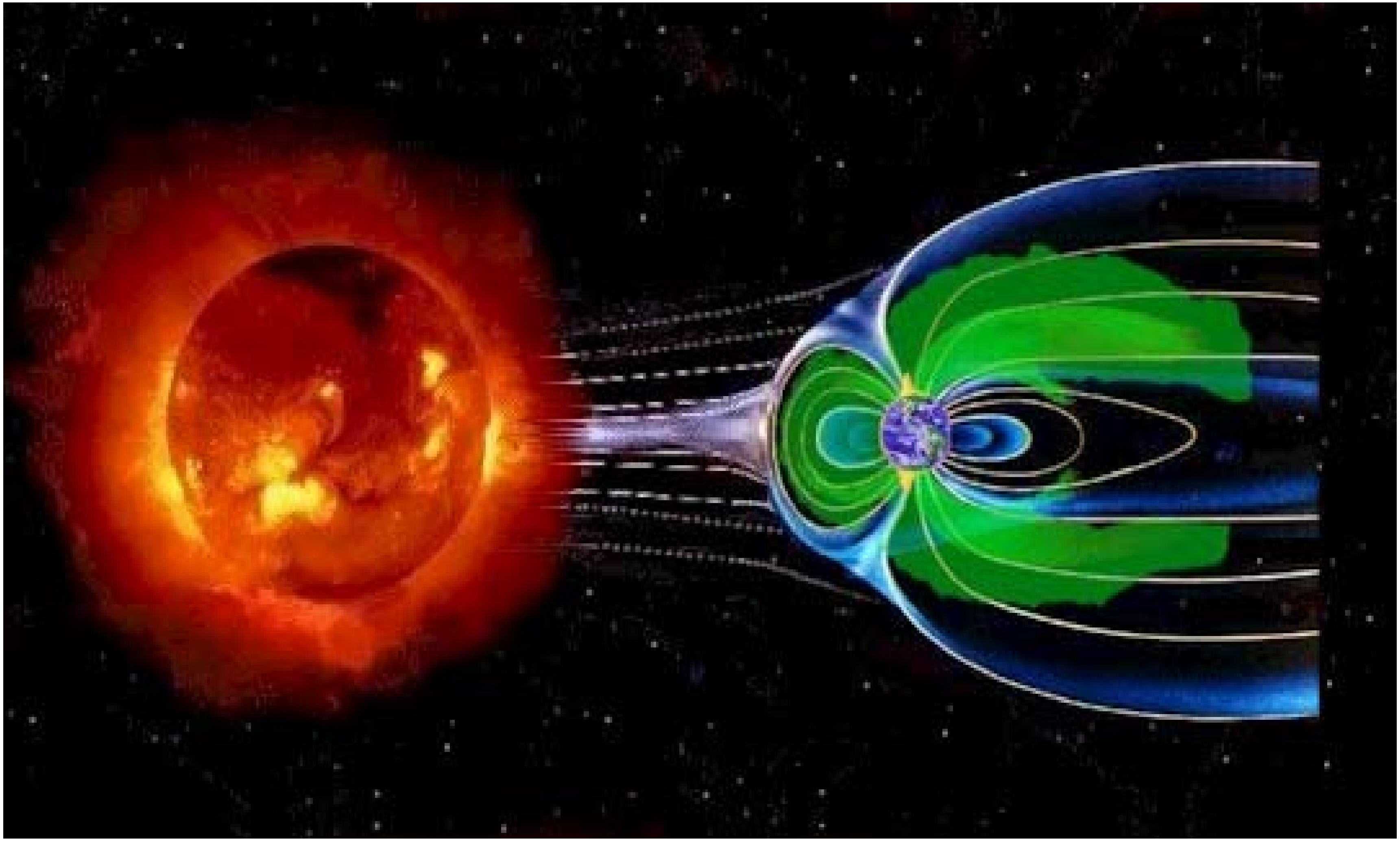
The Community Coordinated Modeling Center

What the CCMC provides:

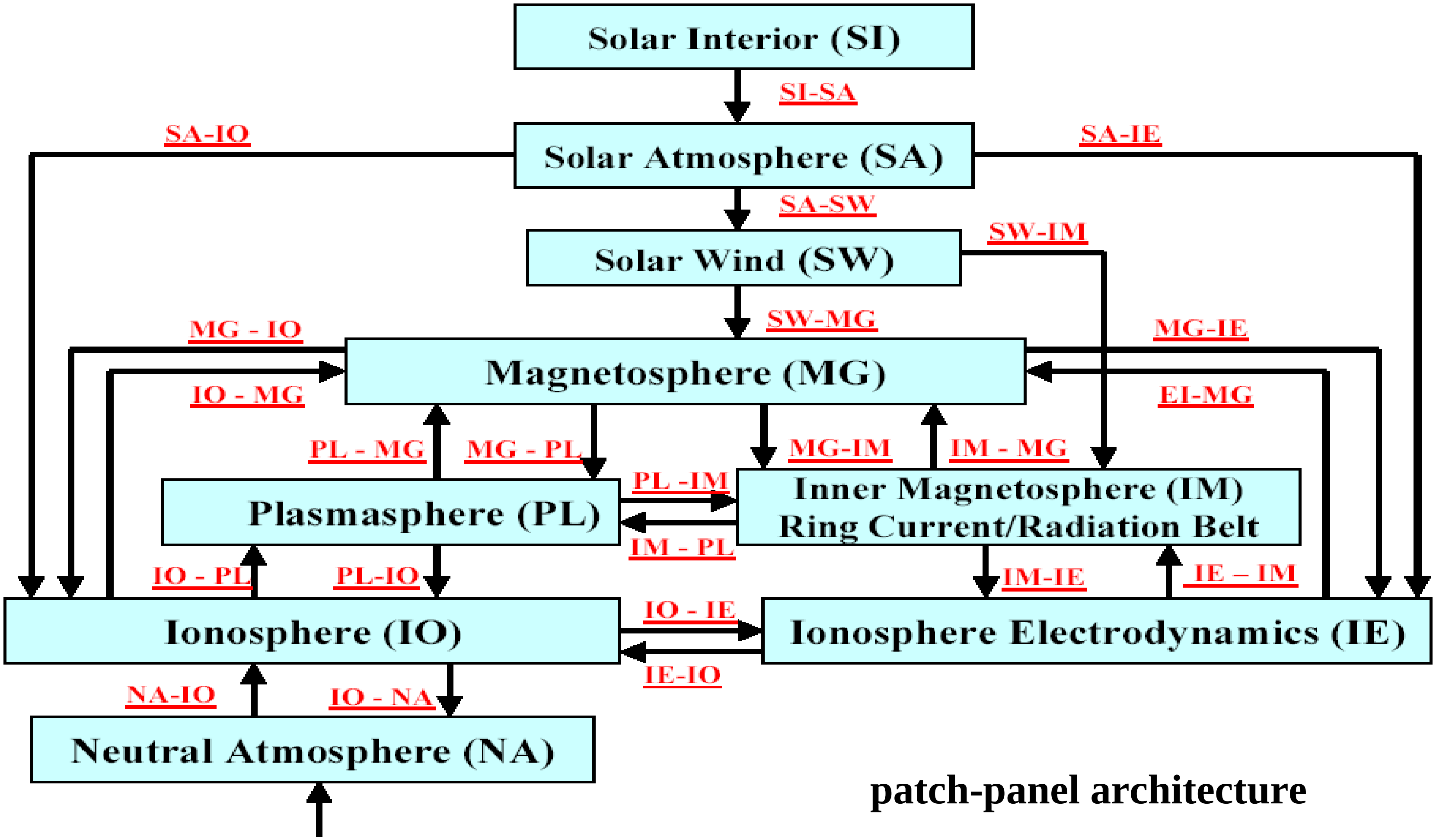
- Scientific validation
- Coupling in collaboration with model owners
- Metrics implementations
- Model runs on request
- Advanced visualization
- **Data Format Standardization**



Covering the Entire Domain

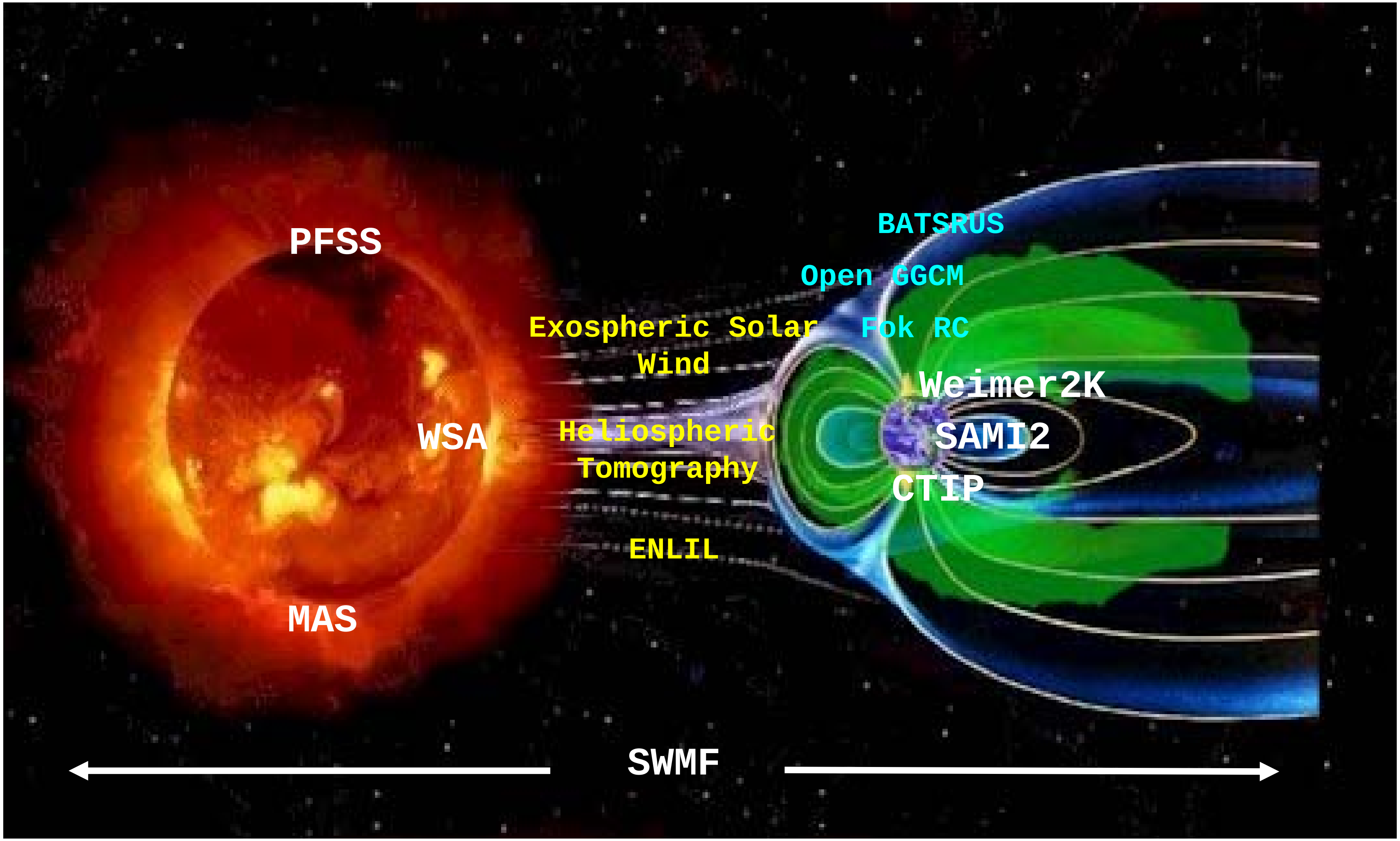


Space Weather Models



patch-panel architecture

Covering the Entire Domain



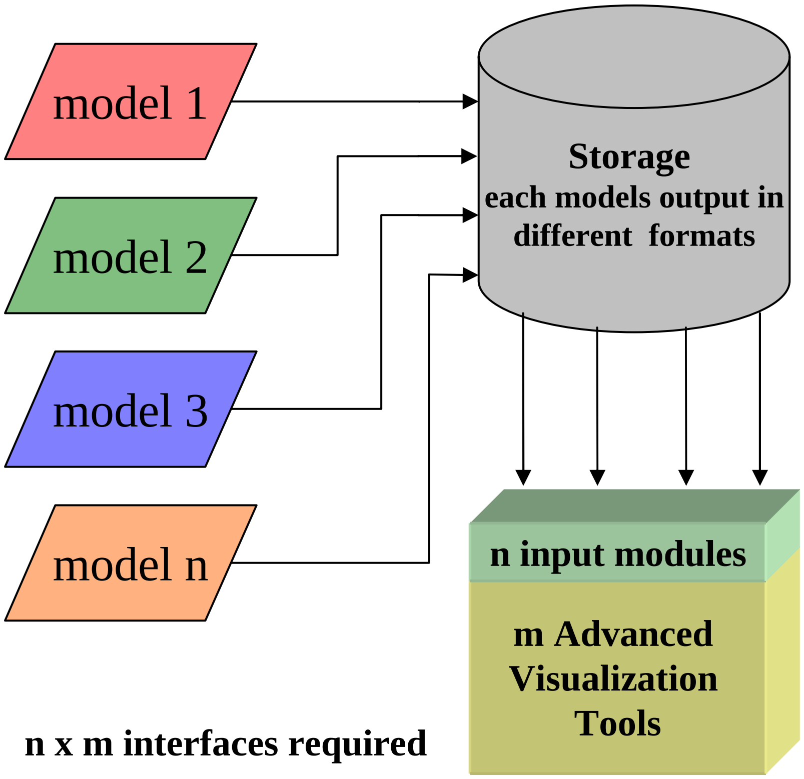
Challenges

- 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

Every Models Output Is Unique

Environment Without Standard

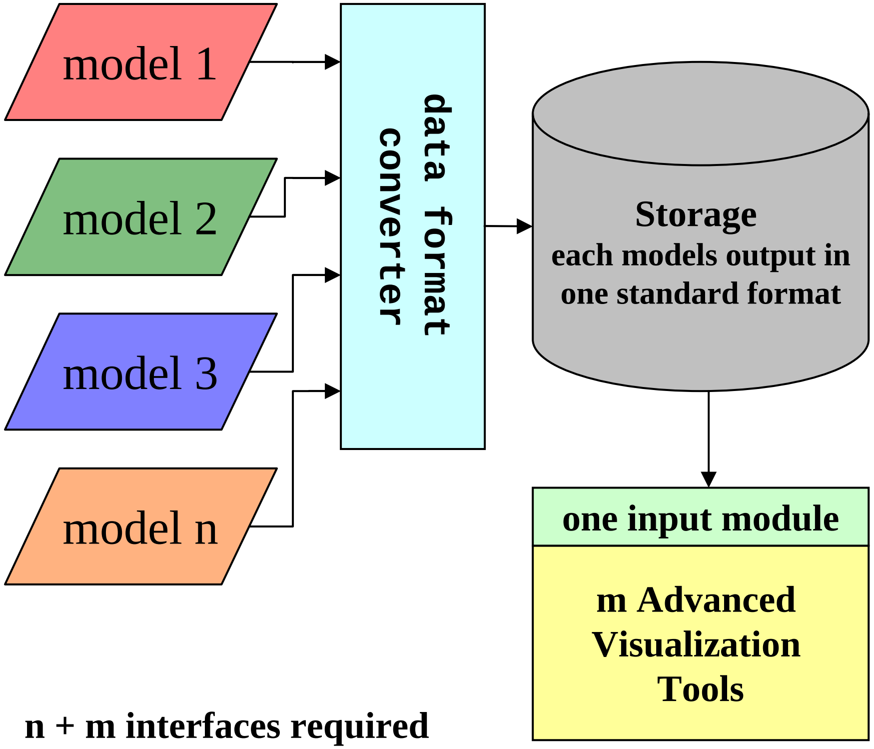
- Specialized I/O routines required for every interface
- Unsuitable for use in flexible model chain
- No commonality between data passing through interfaces



Every Models Output Is Unique

Standardized Environment

- **Original output can be preserved**
- **Standard format for storage, coupling, & visualization**
- **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

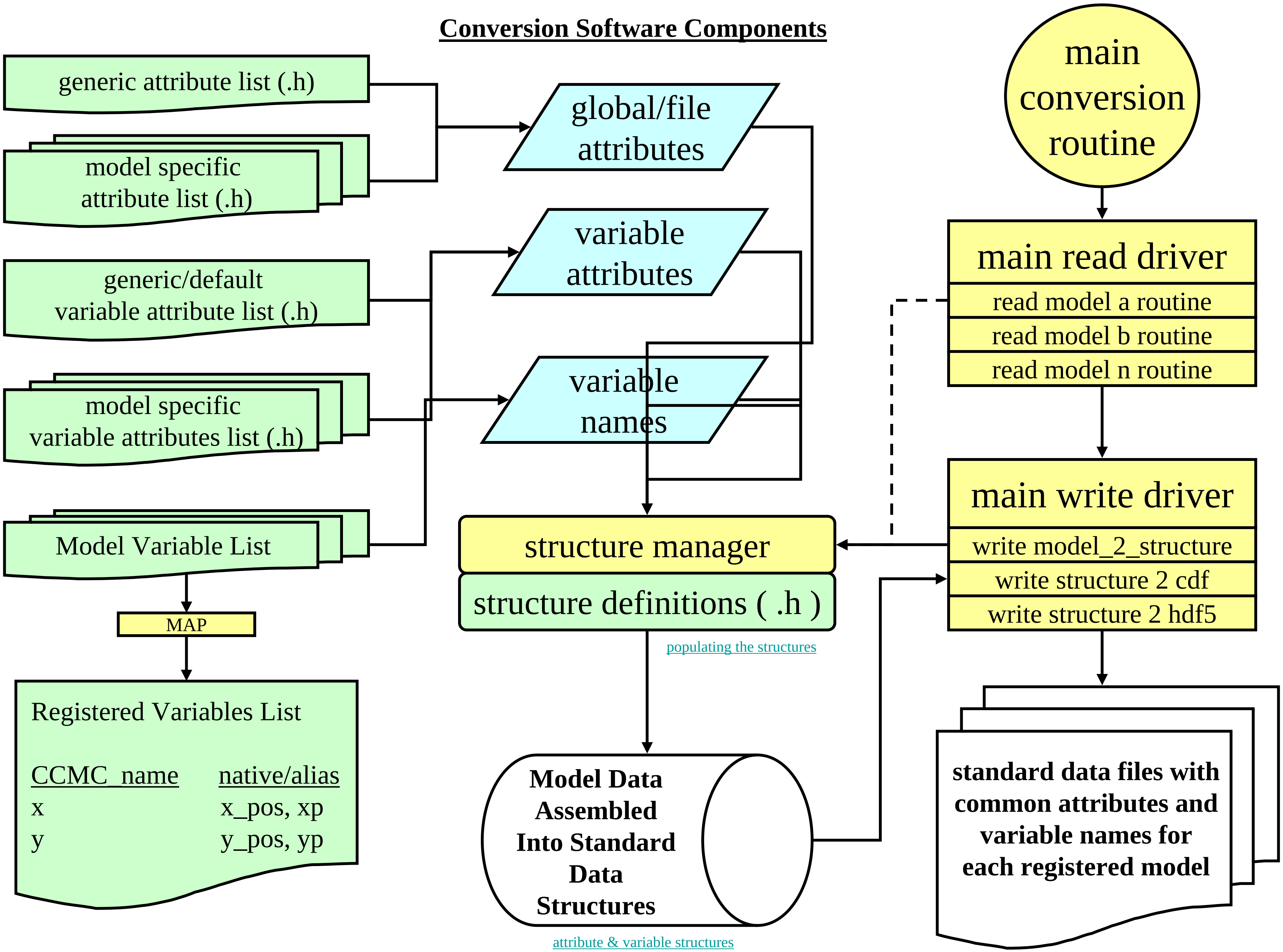
Data Format Standard Options

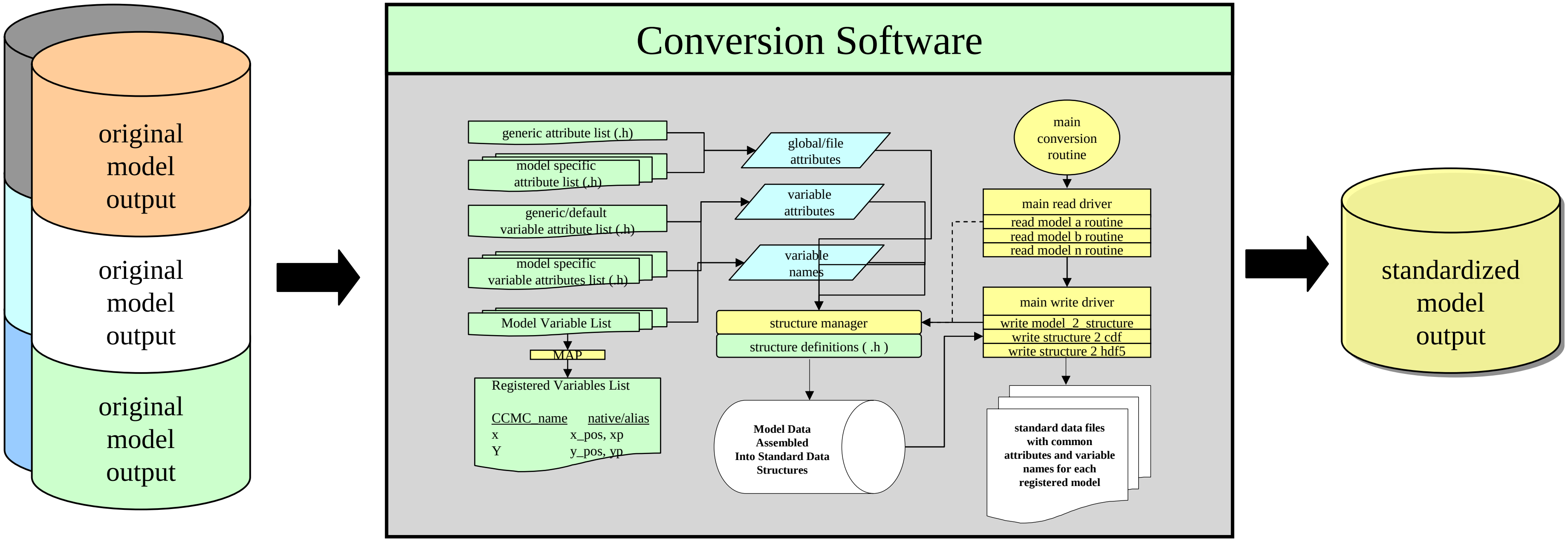
- CDF
- HDF, HDF4, HDF5
- NetCDF
- FITS
- GRIB
- BUFR
- GRADS
- Office Note 29
- Office Note 84
- VICAR
- PDS
- Open Dx Data Model

Metadata

- Aside from the one-to-one data conversion, what additional global & variable meta data do we want to provide?
  - General description of model, howto usage – README
  - Model name and type
  - Date info
    - Run date
    - Generation date
  - Grid Description – # of grids, # of dimensions, dimension size(s)
  - Coordinate system(s)
  - Variable metadata – grid system, min, max, units, description

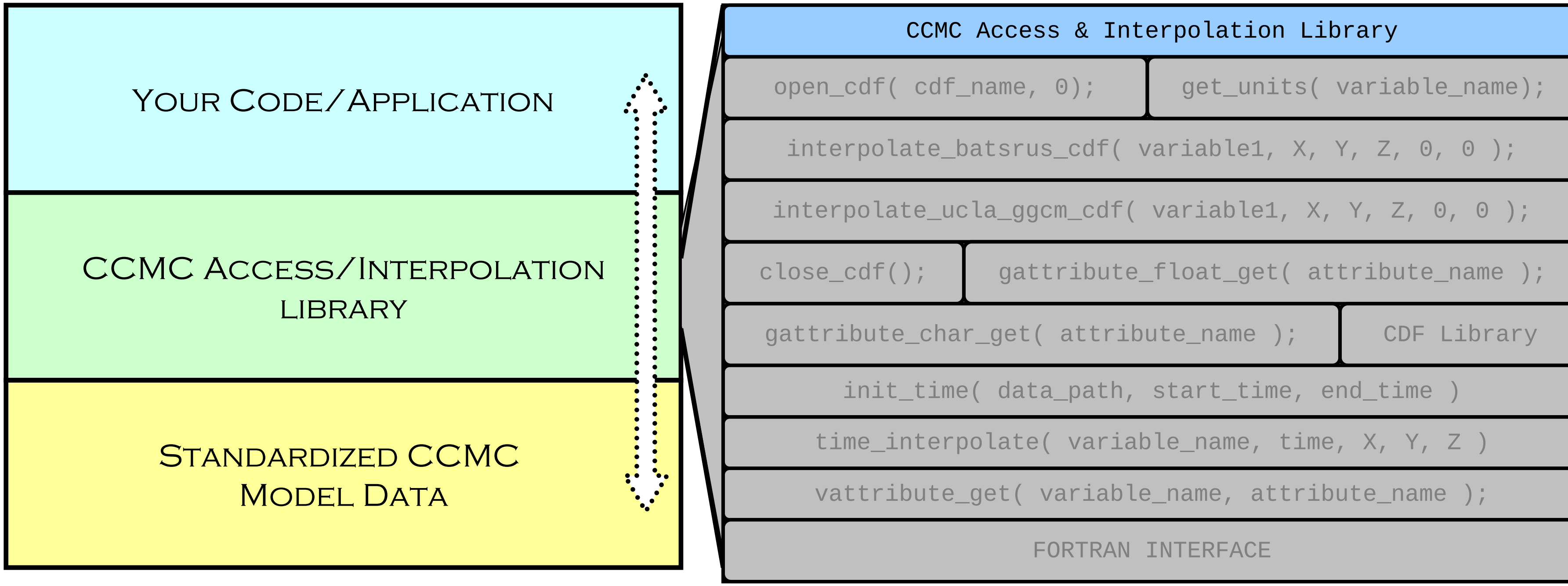
Conversion Software Components




original model output
original model output
original model output
Conversion Software
generic attribute list (.h)
model specific attribute list (.h)
generic/default variable attribute list (.h)
model specific variable attributes list (.h)
Model Variable List
MAP
Registered Variables List
CCMC_name native/alias
x x_pos, xp
Y y_pos, yp
global/file attributes
variable attributes
variable names
structure manager
structure definitions (.h)
Model Data Assembled Into Standard Data Structures
main conversion routine
main read driver
read model a routine
read model b routine
read model n routine
main write driver
write model 2 structure
write structure 2 cdf
write structure 2 hdf5
standard data files with common attributes and variable names for each registered model
standardized model output

Data Access ?

CCMC Access/Interpolation Library



Call from any C supported Programming Language:

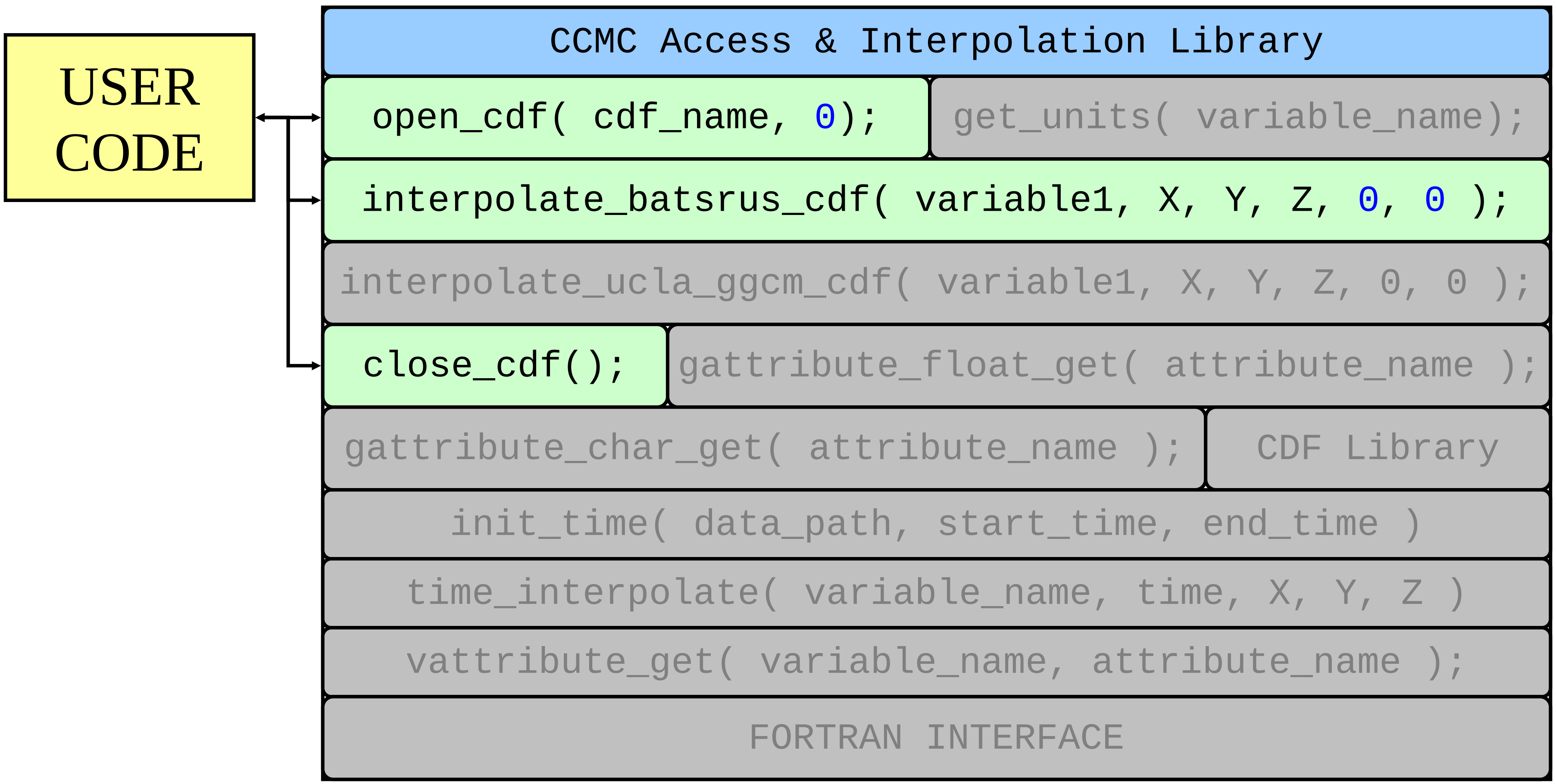
-Fortran	-Java
-C/C++	-Perl
-IDL	-Vtk
-OpenDx	-Your App

Current Standardized Model Output Availability
BATSRUS
OpenGGCM / UCLA-GGCM
CTIP (*Testing Phase*)

Currently Supported Science Data Formats
CDF 2.7
CDF 3.0 (*testing*)
HDF5 (*coming soon*)

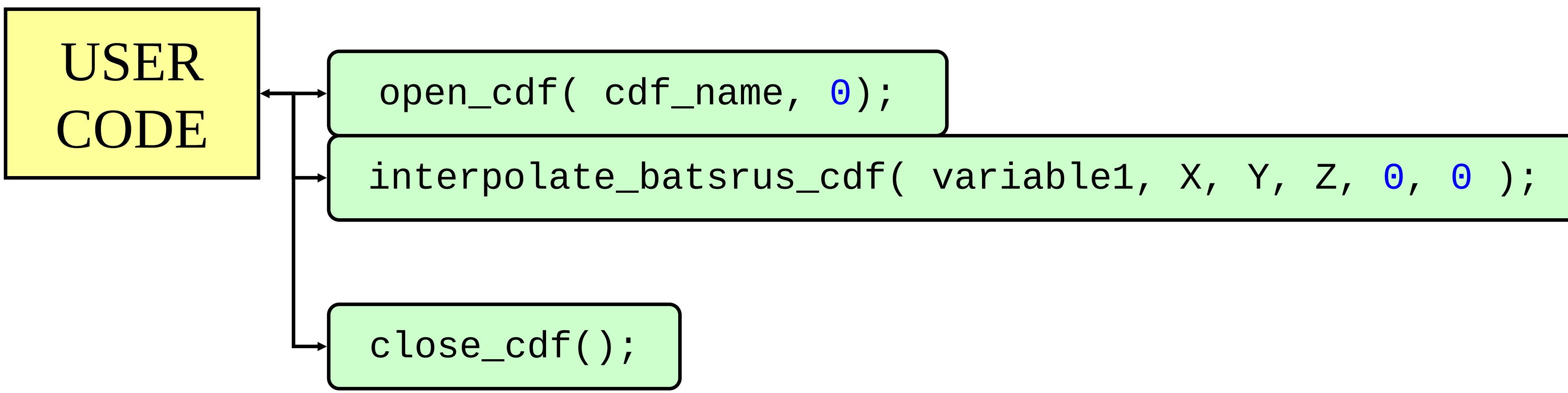
Data Access

- Currently creating a user friendly library of routines that can be called from any C supported language.

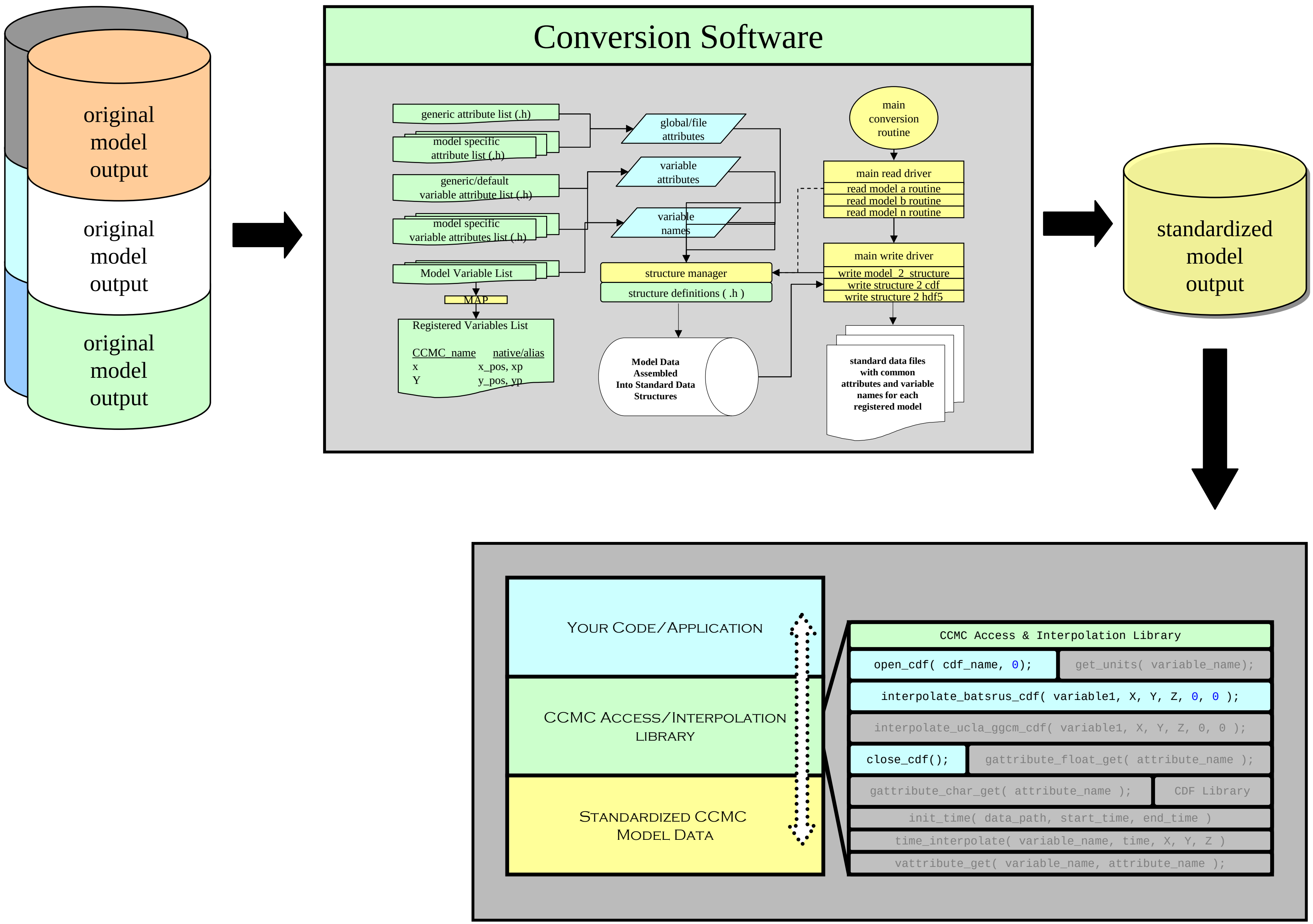


Data Access

- Currently creating a user friendly library of routines that can be called from any C supported language.



- Model output can now be easily accessed using CCMC access/interpolation library and standardized files
  - Allows data to be used in any application that supports the standard c programming language
  - Platform independent

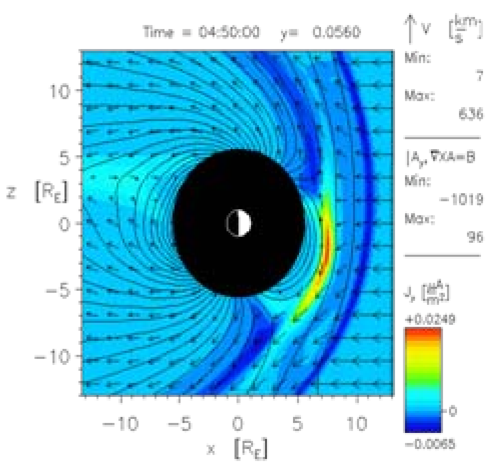

original
model
output
original
model
output
original
model
output
Conversion Software
generic attribute list (.h)
model specific
attribute list (.h)
generic/default
variable attribute list (.h)
model specific
variable attributes list (.h)
Model Variable List
MAP
Registered Variables List
CCMC_name native/alias
x x_pos, xp
Y y_pos, yp
global/file
attributes
variable
attributes
variable
names
structure manager
structure definitions (.h)
Model Data
Assembled
Into Standard Data
Structures
main
conversion
routine
main read driver
read model a routine
read model b routine
read model n routine
main write driver
write model_2_structure
write structure 2 cdf
write structure 2 hdf5
standard data files
with common
attributes and variable
names for each
registered model
standardized
model
output
Your Code/Application
CCMC Access/Interpolation
Library
Standardized CCMC
Model Data
CCMC 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);
close_cdf();
gattribute_float_get(attribute_name);
gattribute_char_get(attribute_name);
CDF Library
init_time(data_path, start_time, end_time)
time_interpolate(variable_name, time, X, Y, Z)
vattribute_get(variable_name, attribute_name);

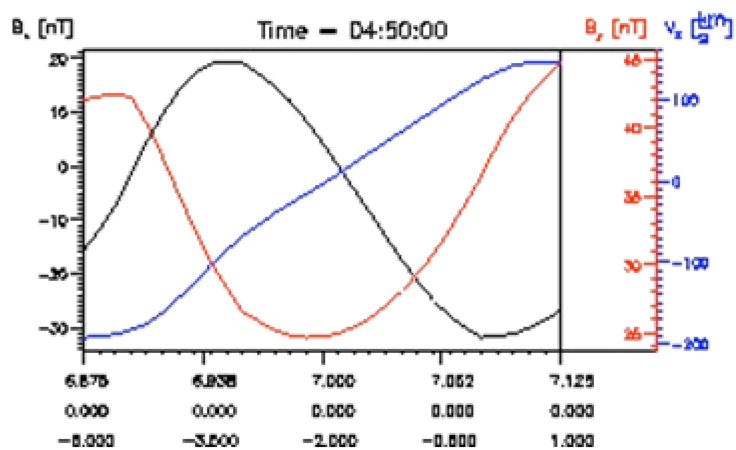
General Usage and Benefits

- Speed and efficiency of direct data access
- Self descriptive data files
- Can be used by anyone with CDF tools & libraries
- Same interface regardless of model
- Platform independent
- Promotes data sharing
- Facilitates code reuse
- CCMC access/interpolation libraries allow model data to be easily integrated into any existing analysis software or application

Specific Usage and Benefits

- Allows CCMC to serve high resolution data that would normally be unavailable
- Integrated Access/Interpolation Libraries into CCMC software infrastructure
    - 3D View Visualization Package (IDL) & Runs-On-Request System
    - Space Weather Explorer Visualization Tool (OpenDx)
- Good for metric studies comparing observation data with simulation data
- Several early adopters
    - Visualization of Kelvin-Helmholtz Waves
    - Particle tracing
- In house research & analysis of high resolution runs





Quick & Efficient Data Access

Analysis only required
Vx, Bx, By in
Range of ~10RE

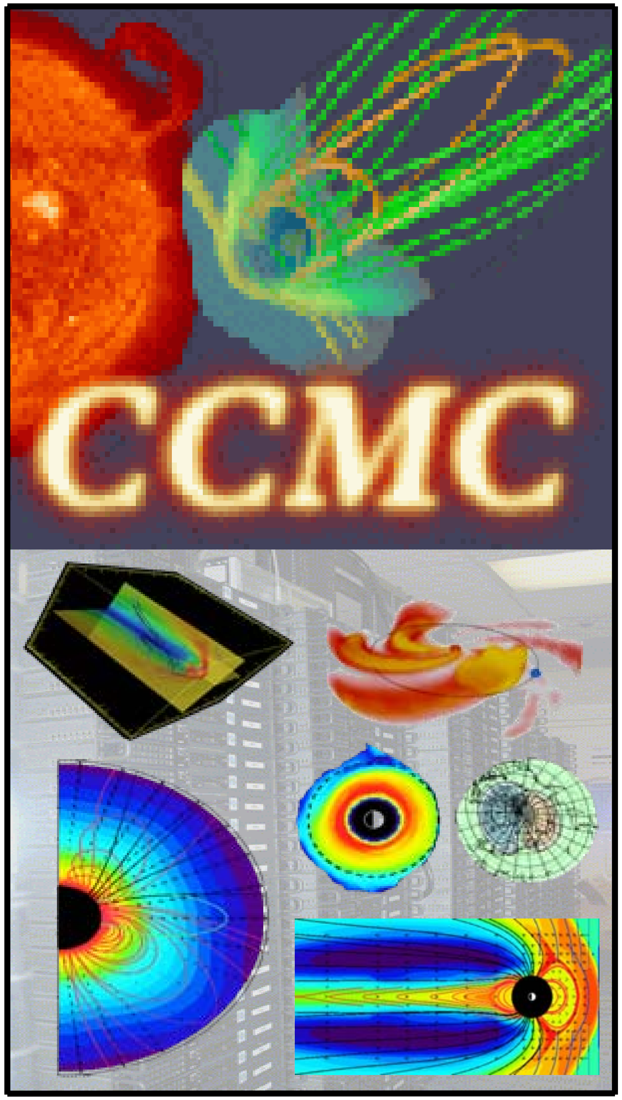
Reduced plot times from
15min to < 5sec

Outstanding Issues

- Variable naming conventions
- How much metadata to pack into each file
- Refining grid description
- Making the transition
- As simulations grow larger keeping original model output may not be feasible
- Add/integrate coordinate system transformation software into existing software suite as well as specific data analysis tools

Summary

- General grid description scheme in place
- Clearly defined set of core metadata elements
- Structure oriented architecture ensures flexibility
- Conversion software currently supports:
  - BATSRUS, BATSRUS SWMF Framework, UCLA-GGCM/OpenGGCM, & CTIP
- CCMC access/interpolation library currently supports:
  - BATSRUS, BATSRUS SWMF Framework, UCLA-GGCM/OpenGGCM CDF files
  - Interface to easily extract global & variable metadata
  - Time interpolation for entire data sets
  - Fortran Interface
- Add Solar & Ionosphere Models to software - CTIP testing phase
- Implement HDF 5 conversion module
- Library is currently available for use and we encourage feedback - questions, comments, and/or suggestions



Data Format Standardization of Space Weather Model Output at The Community Coordinated Modeling Center

M. Maddox

NASA Goddard Space Flight Center

http://ccmc.gsfc.nasa.gov

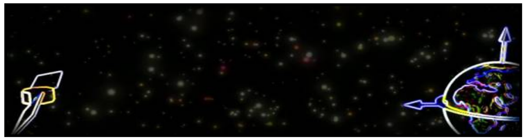
Hilton
Clearwater Beach Resort

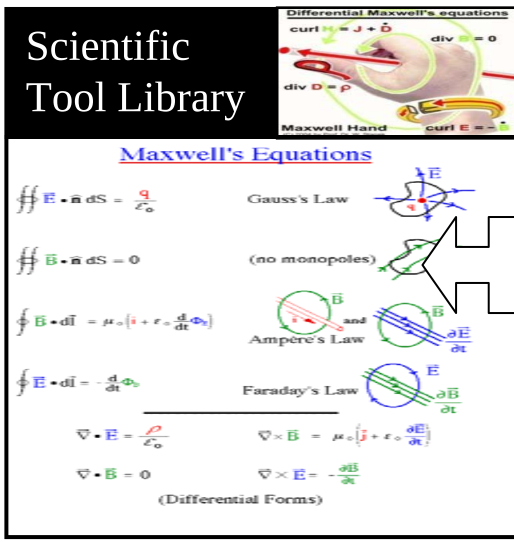
CCMC Workshop

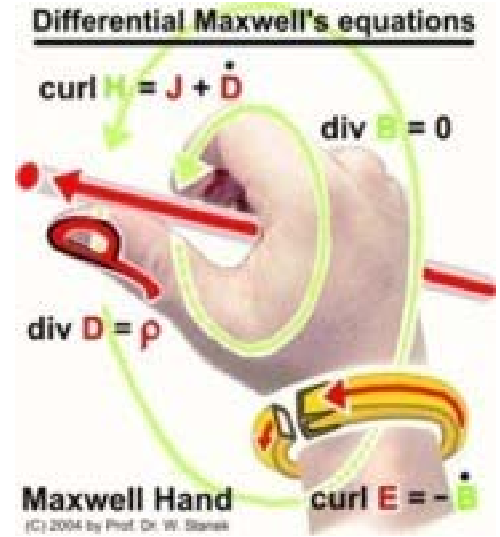
October 11-14, 2005

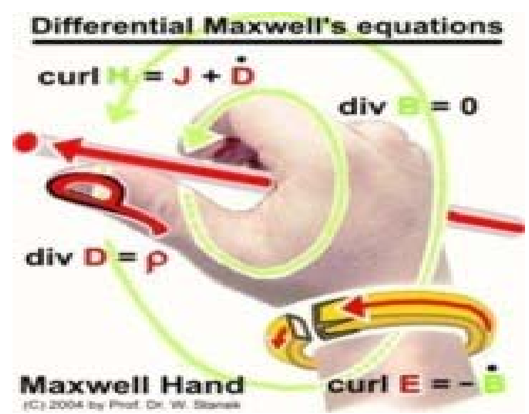
Tuesday, October 11th PM Session:

Frameworks and Data infrastructure




Scientific
Tool Library
Maxwell's Equations
Gauss's Law
(no monopoles)
Ampère's Law
Faraday's Law
(Differential Forms)


Differential Maxwell's equations
curl H = J + Ḋ
div B = 0
div D = ρ
Maxwell Hand
curl E = - Ḃ
(C) 2004 by Prof. Dr. W. Stanek


Differential Maxwell's equations
curl H = J + Ḋ
div B = 0
div D = ρ
Maxwell Hand
curl E = - Ḃ
(C) 2004 by Prof. Dr. W. Stanek

Supplemental Slides

Attribute	Description	Status
README	General description describing the model, contents of CDF, and HOWTO usage.	
README_visualization	Guidelines for visualizing data contained in file.	
model_name	Name of the registered model that produced the data	
model_type	The type of model used to produce data i.e. Global Magnetosphere	
generation_date	Date of generation or run date…	
original_output_file_name	Name of the original model data file that was converted to current CDF file	
run_registration_number	CCMC Runs on Request registration number for runs submitted through online system	
generated_by	Personal identifying info (First Name Last Name)	
terms_of_usage	*For tracking purposes for our government sponsors, we ask that you notify the CCMC whenever you use CCMC results in a scientific publication or presentation.*	
grid_system_count	The number n of how many grid systems are used and/or described in the current cdf file NOTE: If n > 1 the additional grid attributes will be defined in corresponding model_attributes.h file	
grid_system_n_number_of_dimensions	The number m of how many dimensions are in grid n. So for every grid there will be a corresponding grid_system_n_number_of_dimensions attribute i.e. The first grid will have an attribute grid_system_1_number_of_dimensions	
grid_system_n_dimension_m_size	Size of dimension m for grid n	
grid_system_n	Outline how particular grid system is defined by showing coordinates used ie. [X, Y, Z] were X,Y,Z are position variables defined in current CDF.	
output_type	Define the type of output is contained in CDF file. i.e. Global Magnetosphere model with Ionosphere output	
standard_grid_target	Defines a standard target grid and coordinate system for which the current models output can be converted to using an external coordinate transformation package.	
grid_n_type	Keywords identifying all grids used in current model output. Grid types will be registered in external coordinate transformation package.	
start_time	Time in CDF Epoch3 format (YYYY-MM-DDThh:mm:ss.msecZ) signifying beginning of the simulation	
end_time	Time in CDF Epoch3 format (YYYY-MM-DDThh:mm:ss.msecZ) signifying end of the simulation	
run_type	An event or model extracted from DatabaseInfo file	

CCMC Global Attributes

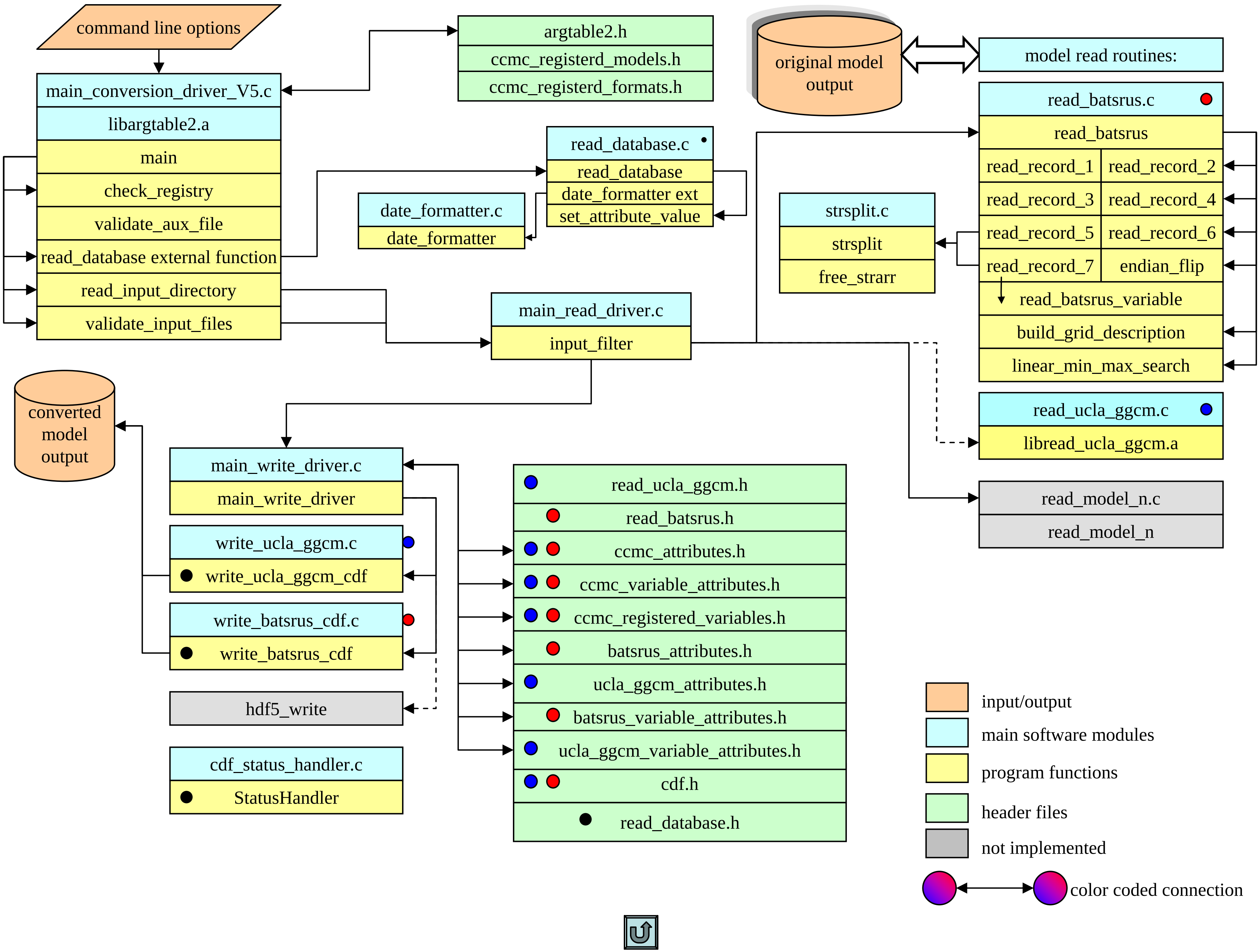
- README
- README_visualization
- model_name
- model_type
- Generation_date
- Original_output_file_name
- Run_registration_number
- Generated_by
- Terms_of_usage
- Grid_system_count
- Grid_system_n_number_of_dimensions
- Grid_system_n_dimension_m_size
- Grid_system_n
- Output_type
- Standard_grid_target
- Grid_n_type
- Start_time
- End_time
- Run_type

Model Specific Global Attributes

•••

•••

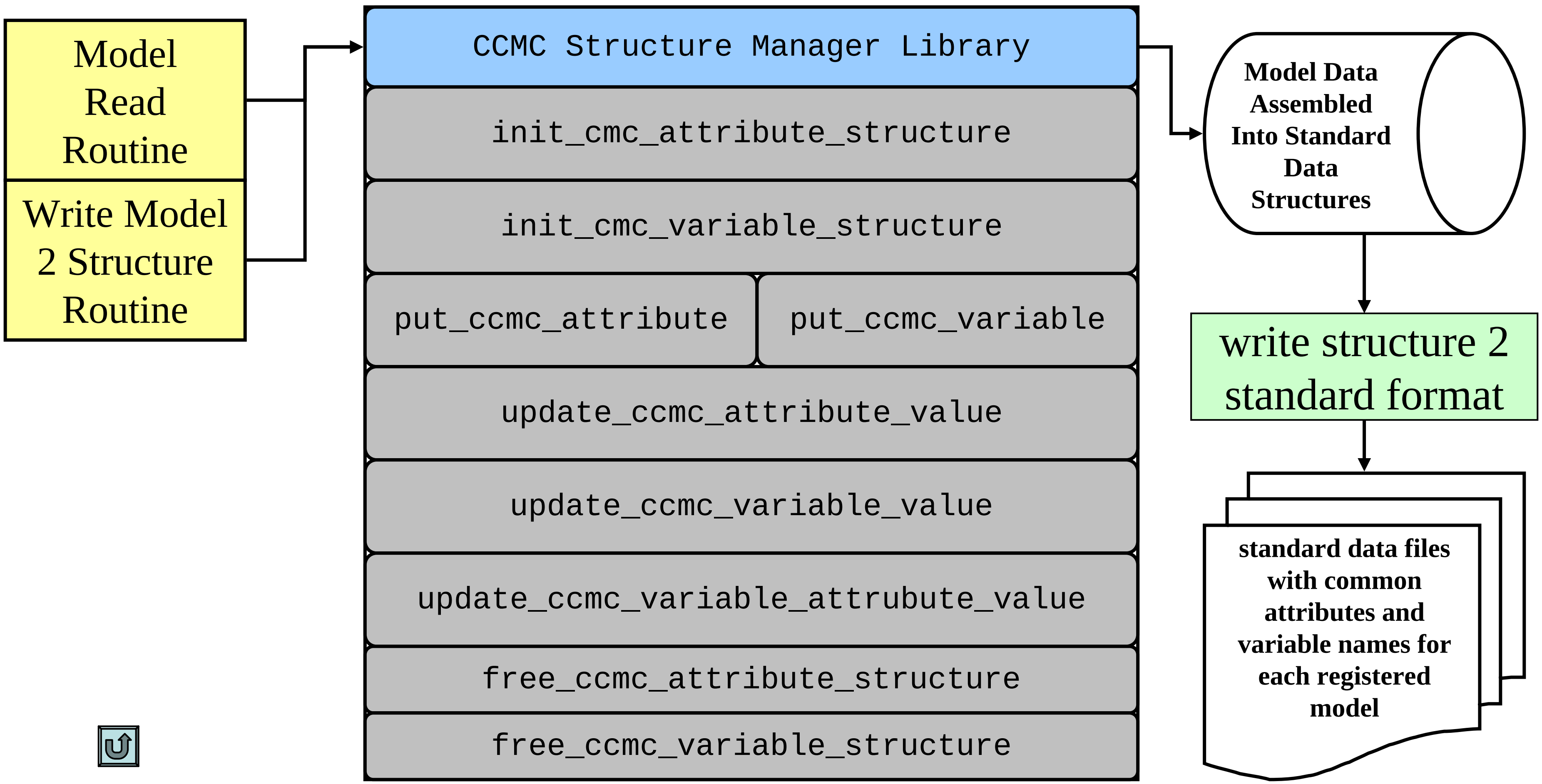
Attribute	Description	Status
valid_min	Smallest valid value for a particular variable. If actual value is less than this valid_min, the actual value is physically impossible and/or was generated in error	
valid_max	Largest valid value for a particular variable. If actual value is greater than this valid_max, the actual value is physically impossible and/or was generated in error	
units	The particular units of measurement for a variable	
grid_system	The grid system in which a variable is on	
mask	Mask value	
desciption	A description of the variable	
is_vector_component	Boolean value identifies if variable is a vector component or scalar variable	
position_grid_system	…	
data_grid_system	…	
actual_min	The smallest value for a particular variable	
actual_max	The largest value for a particular variable	



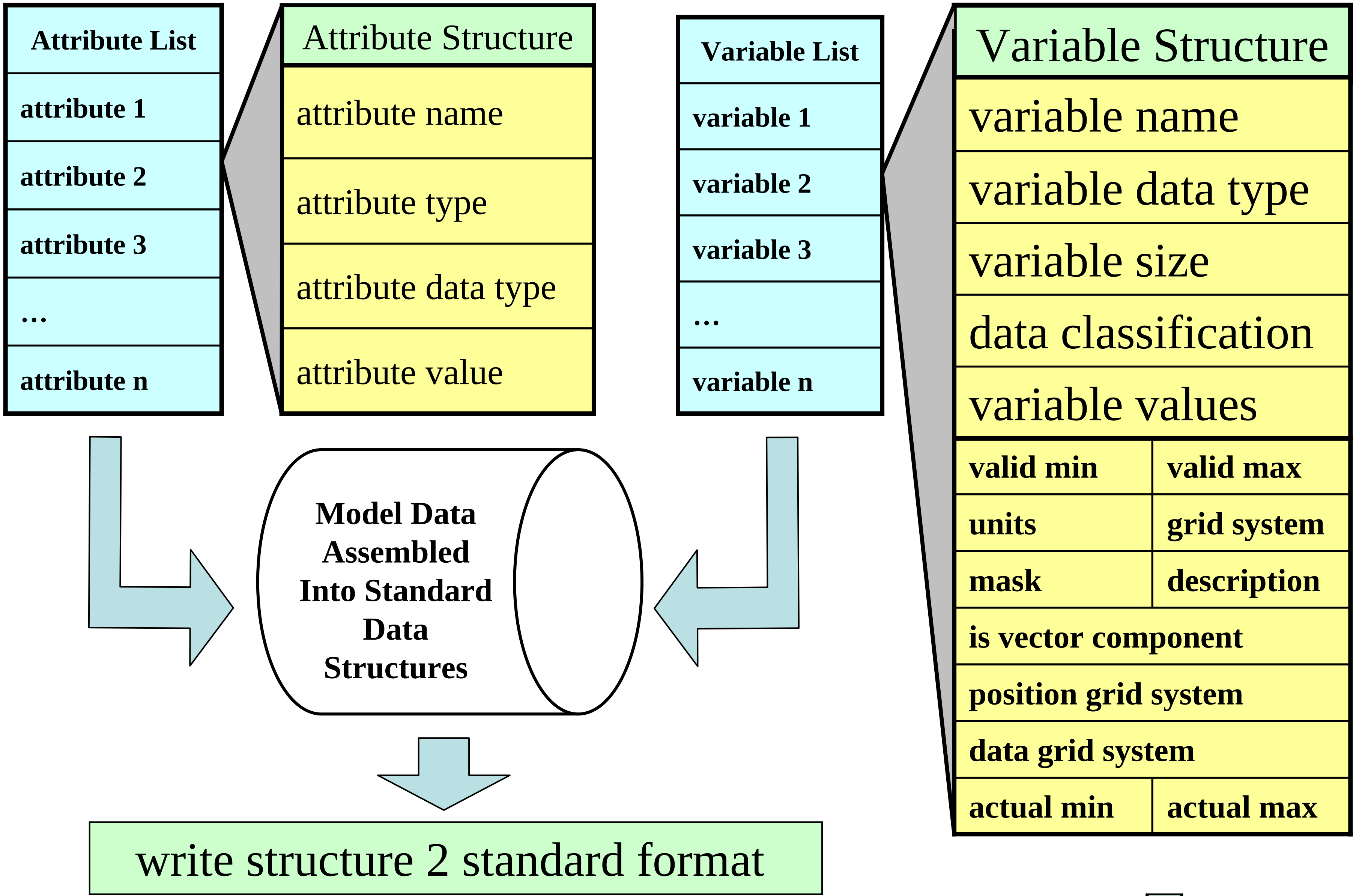
command line options
main_conversion_driver_V5.c
libargtable2.a
main
check_registry
validate_aux_file
read_database external function
read_input_directory
validate_input_files
argtable2.h
ccmc_registerd_models.h
ccmc_registerd_formats.h
original model output
model read routines:
read_batsrus.c
read_batsrus
read_record_1
read_record_2
read_record_3
read_record_4
read_record_5
read_record_6
read_record_7
endian_flip
read_batsrus_variable
build_grid_description
linear_min_max_search
read_database.c
read_database
date_formatter ext
set_attribute_value
date_formatter.c
date_formatter
strsplit.c
strsplit
free_strarr
main_read_driver.c
input_filter
read_ucla_ggcm.c
libread_ucla_ggcm.a
read_model_n.c
read_model_n
converted model output
main_write_driver.c
main_write_driver
write_ucla_ggcm.c
write_ucla_ggcm_cdf
write_batsrus_cdf.c
write_batsrus_cdf
hdf5_write
cdf_status_handler.c
StatusHandler
read_ucla_ggcm.h
read_batsrus.h
ccmc_attributes.h
ccmc_variable_attributes.h
ccmc_registered_variables.h
batsrus_attributes.h
ucla_ggcm_attributes.h
batsrus_variable_attributes.h
ucla_ggcm_variable_attributes.h
cdf.h
read_database.h
input/output
main software modules
program functions
header files
not implemented
color coded connection

Populating the Structures

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



Standardized Attribute & Variable Structure Lists

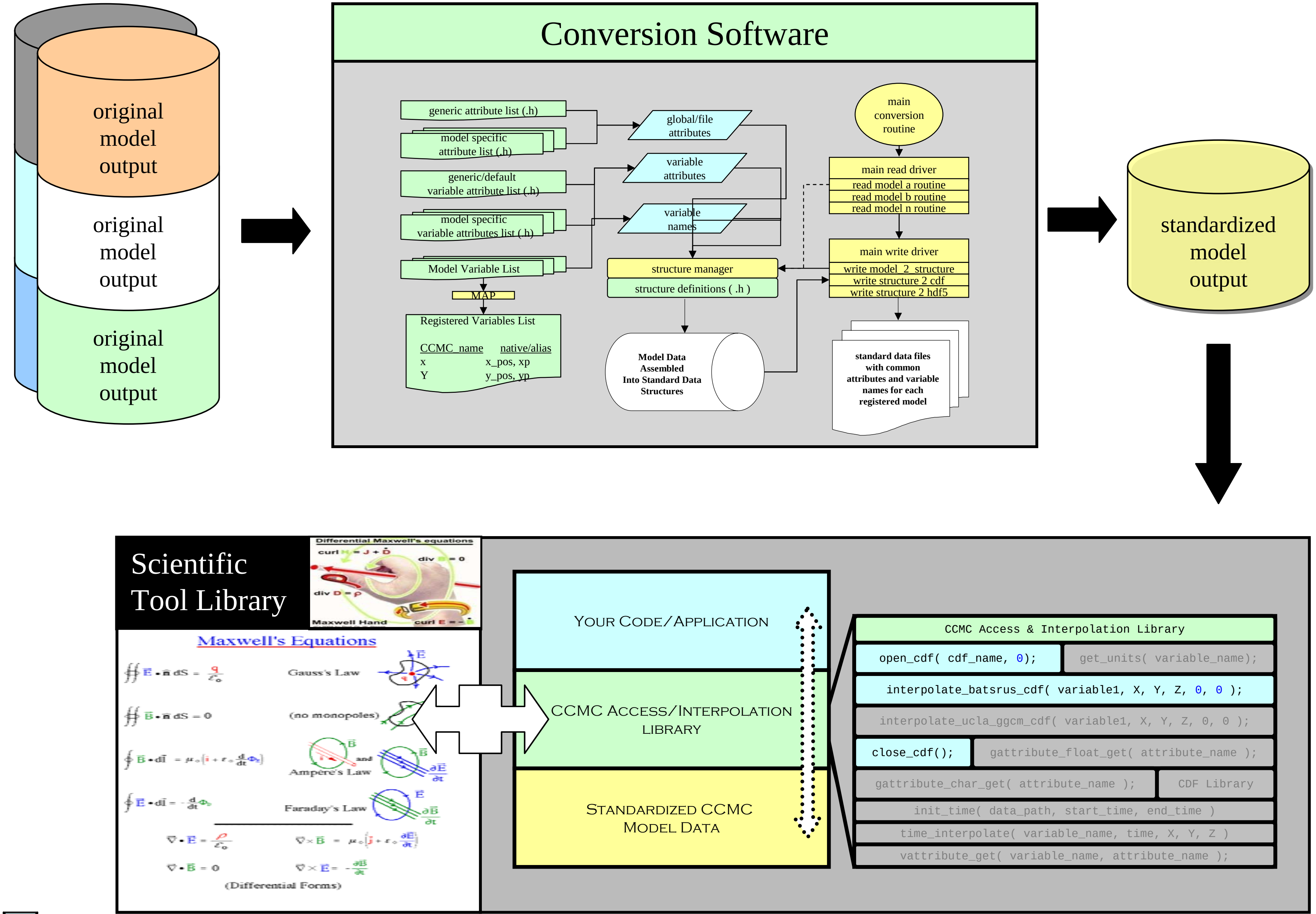


Standardization is Feasible
(Summary from CCMC Workshop 2003)

- BATRUS .Out to CDF conversion results promising
  - 1.5 second uncompressed CDF creation time
  - Resulting file size virtually unchanged
- OpenDx successfully imported CDF data using standard input module *(only had to specify input file name)*
  - Requires minimal initial development to correctly categorize imported data
- Working toward developing and implementing a flexible data format standardization software tool within the CCMC

Next Steps (from 2003 CCMC Workshop)

- Develop standard grid description scheme
- Implement HDF 5 conversion module
- Test BATRUS output conversion performance with HDF 5 data standard
- Compare CDF vs. HDF 5 performance
- Decide use of either CDF or HDF5 or both
- Develop standard “in-house” naming conventions for variables


original model output
original model output
original model output
Conversion Software
generic attribute list (.h)
model specific attribute list (.h)
generic/default variable attribute list (.h)
model specific variable attributes list (.h)
Model Variable List
MAP
Registered Variables List
CCMC_name native/alias
x x_pos, xp
Y y_pos, yp
global/file attributes
variable attributes
variable names
structure manager
structure definitions (.h)
Model Data Assembled Into Standard Data Structures
main conversion routine
main read driver
read model a routine
read model b routine
read model n routine
main write driver
write model 2 structure
write structure 2 cdf
write structure 2 hdf5
standard data files with common attributes and variable names for each registered model
standardized model output
Scientific Tool Library
Differential Maxwell's equations
Maxwell Hand
Maxwell's Equations
Gauss's Law
(no monopoles)
Ampere's Law
Faraday's Law
(Differential Forms)
YOUR CODE/APPLICATION
CCMC ACCESS/INTERPOLATION LIBRARY
STANDARDIZED CCMC MODEL DATA
CCMC 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);
close_cdf();
gattribute_float_get(attribute_name);
gattribute_char_get(attribute_name);
CDF Library
init_time(data_path, start_time, end_time)
time_interpolate(variable_name, time, X, Y, Z)
vattribute_get(variable_name, attribute_name);