The Flexible Climate Data Analysis Tools (CDAT) for Multi-model Climate Simulation Data

Dean N. Williams
PCMDI/LLNL
7000 East Ave.
1 (925) 423-0145
williams13@llnl.gov

Charles M. Doutriaux
PCMDI/LLNL
7000 East Ave
1 (925) 422-1487
doutriaux1@llnl.gov

Robert S. Drach
PCMDI/LLNL
7000 East Ave
1 (925) 422-8303
drach1@llnl.gov

Renata B. McCoy
PCMDI/LLNL
7000 East Ave
1 (925) 422-8303
mccoy20@llnl.gov

Abstract

Being able to incorporate, inspect, and analyze data with newly developed technologies, diagnostics, and visualizations in an easy and flexible way has been a longstanding challenge for scientists interested in understanding the intrinsic and extrinsic empirical assessment of multi-model climate output. To improve research ability and productivity, these technologies and tool advancements must be made easily available to help scientists understand and solve complex scientific climate changes.

To improve productivity and ease the challenges of incorporating new tools into the hands of scientists, the Program for Climate Model Diagnosis and Intercomparison (PCMDI) is developing the Climate Data Analysis Tools (CDAT). CDAT is an application for developing and bringing together disparate software tools for the discovery, examination, and intercomparison of coupled multi-model climate data. By collaborating with top climate institutions, computational organizations, and other science communities, the CDAT community of developers is leading the way to provide proven data management, analysis, visualization, and diagnostics capabilities to scientists. This communitywide effort has developed CDAT into a powerful and insightful application for knowledge discovery of observed and simulation climate data.

As the analysis engine in the Earth System Grid (ESG) data infrastructure, CDAT is making it possible to remotely access and analyze climate data located at multiple sites around the world.

1. INTRODUCTION

1.1 Project Review

The Program for Climate Model Diagnosis and Intercomparison (PCMDI) (Gates et al., 1999) originally developed the Climate Data Analysis Tools (CDAT) (Williams, 1997) under the funding of the Department of Energy's Office of Biological and Environmental Research at the Lawrence Livermore National Laboratory in Livermore, California.

In its conception, CDAT was originally designed to provide basic capabilities needed for validating, comparing, and diagnosing climate model behavior. Driven by the need to acquire and share resources (i.e., worldwide collaboration), CDAT expanded its software and application domain to an open-source environment. In this open-source software environment, external developers were allowed to contribute greatly to its advancement and deliver software products on par with developers at PCMDI.

In over a decade of development, CDAT has advanced the data management infrastructure and analysis supporting very large data volumes generated by Model Intercomparison Projects (MIPs) and observational programs that are widely dispersed among many international institutions. Separately and as part of the Earth System Grid (ESG) infrastructure (Williams et al., 2009), CDAT has been instrumental in the archival and analysis of several large climate data experiments. The best known of these is the Climate Model Intercomparison Project, Phase 3 (CMIP3), which is a collection of multi-model climate data used to produce the Intergovernmental Panel on Climate Change (IPCC) Fourth Assessment Report (AR4).

Adhering to common data formats and metadata standards, CDAT is preparing for future MIP experiments, including the Climate Model Intercomparison Project, Phase 5 (CMIP5) Intergovernmental Panel on Climate Change (IPCC) Fifth Assessment Report (AR5) archive.

1.2 Scientific and Computational Challenges (Motivation)

The expanding CDAT community continues to adapt and meet the diverse scientific and computational challenges faced by the climate change. Its primary objective is to develop and use existing advanced software designed to disseminate and diagnose multi-model climate data vital to understanding climate change. This interconnection of disparate software into a seamless infrastructure enables scientists to handle and analyze ever-

increasing amounts of data and enhances their research by eliminating the need to write numerous large proprietary programs.

This "glue–like" architecture has many advantages. To begin, it allows data to be entered into the system where it becomes accessible by all other software components. Furthermore, it allows scientists to have adequate control over fundamental pieces of the climate change diagnosis process to ensure that the targeted outcome is of high quality and complete. These advantages point to a pressing need for the community to expand and develop free software that is built for sharing.

1.3 Vision

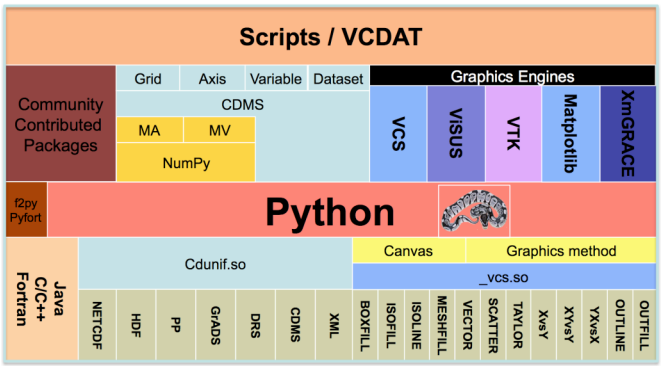

Figure 1: CDAT architectural layer.

The vision of CDAT has always been to leverage off the community to provide efficient resources such as storage, diagnosis, and visualization of climate model and observational data through the investment of past, present, and future software activities. Based on the Python (Lutz, 2007) open-source environment, CDAT has achieved worldwide collaboration in its use and in its integration and development of Python, Java, C/C++, and FORTRAN codes. (See Figure 1.)

Over the coming months, CDAT will leverage off the Earth System Grid infrastructure to provide distributed products to a growing list of customers. As its software base continues to expand and grow, it will continue to be important to the climate community in the development, testing, validation, and intercomparison of global climate models.

1.4 Scientific Impact

Climate change research is a model- and data-driven process. There is no research without data and no useable data without a useful tool that can ingest, manipulate, and assist in the diagnosis of that data for study. Since the mid 1990s, CDAT has been accessible, in one form or another, for the IPCC assessment reports—each one asserting a stronger anthropogenic case for the cause of global warming. For the reports, CDAT was used to process and analyze data, and to produce publication-quality images. For other MIPs, CDAT has been used in the same capacity, providing scientists with the ability to access, examine, and interpret their unique datasets for climate assessment.

1.5 Building on the Community

As CDAT quickly evolved, a much deeper and richer set of tools were developed by the community to help facilitate future climate predictions using state-of-the-art analysis, diagnosis, and visualizations. This was attainable by the use of the powerful Python infrastructure, which allows for the "gluing" of disparate software components. To begin with, the developers of CDAT packaged legacy community climate FORTRAN codes in Python wrappers. Community codes outside the field of climate were also sought after and integrated into CDAT. This type of community development, mixing existing climate, Python, and other scientific application software seamlessly allows CDAT to be flexible and adjust quickly to community needs. Because of this community development in Python, CDAT is able to have fast code growth without requiring changes to the underlying fundamental infrastructure.

1.6 Project Partners

Project partners are for the most part unknowing participants in the development of CDAT. A large part of CDAT is leveraged under the Python consortium. What distinguish CDAT from Python are the specific climate-related packages. A significant number of software contributions to CDAT are from the climate community of which most is written in FORTRAN. CDAT partners are diverse and range from a variety of climate and forecast endeavors. Although most partners are developers with heavy computational backgrounds, an increasing number of contributors are scientists providing scripts and diagnostics to aid in understanding atmospheric phenomena.

1.7 Project Timeline

The current version of CDAT (version 5.1) was released in May 2009. The next major release of CDAT (version 6.0) is scheduled for release in Spring 2010. This next major release will contain a new graphical user interface along with more diagnostics and 3D visualization. Major overhauls have been made to the underlying numeric package and to the *de facto* standard 1D and 2D visualization package.

1.8 Conclusion

The principal mission of CDAT is to develop improved and seamless interconnected methods for storage, diagnosis, validation, and intercomparison of multi-model climate simulation data. It is designed to aid climate scientists in the study of climate model output. Its concept is simple and flexible enough to interchange its components and expand for future needs. Although it has been designed primarily for the climate community, it has been enhanced and used to meet the needs of other geophysical sciences.

2. HANDLING AND MANAGEMENT OF DATA

2.1 Data and Management Challenges

For CMIP5, the archive is estimated to be between 6 to 10 petabytes. A centralized repository of a core subset of

the CMIP5 archive will be housed at PCMDI and estimated to be between 600 terabytes to 1 petabyte. (This archive will be the largest distributed multi-model climate archive ever and indeed the largest collection of multi-model climate data ever assembled in one place.) To accommodate this archive, technologies are underway to revolutionize how these data are hosted, managed, searched, disseminated, and analyzed on such a large scale. Although not the primary focus of this paper, the ESG will uniquely harvest CMIP5 data from the world's leading climate data centers and allow researchers to search and access data at an unprecedented rate (Williams et al., 2009). To this end, ESG is the starting point from which all data will be processed, cataloged, discovered, retrieved, and shared. It is in this larger ESG enterprise system that CDAT will be used to process, manipulate, and visualize data where it resides.

2.2 Data formats

Data is a critical part of any research. In climate, it is defined in two parts: resulting information (i.e., data), and information describing what is to be done with the resulting information and how to use it (i.e., metadata). These distinct pieces of information are usually stored in a certain way known as a format. In the climate modeling community, more and more groups are opting to store their data in the network Common Data Form (netCDF) (UCAR, 2008). In addition to netCDF there are other popular data formats to choose from such as the Hierarchical Data Format (HDF) (HDF Group, 2009) and the GRIdded Binary (GRIB) (WMO, 2009) format. For the CMIP5 archive, the data will be stored in the netCDF format.

2.3 Climate and Forecast (CF) Convention

Along with data formats, the climate community has selected a *de facto* methodology for defining the metadata that provides a definitive description of what the data in each variable represents, and the spatial and temporal properties of the data. Combined with the netCDF API, the Climate and Forecast (CF) (CF, 2009) metadata convention by definition enables users of data from different sources to decide which quantities are comparable, and facilitates building applications with powerful extraction, regridding, and display capabilities. This metadata convention makes it possible for other georeferencing data such as atmospheric, ocean, atmospheric chemistry, biogeochemistry, satellite observations, and others to be compared and displayed together with little or no effort on the part of scientists.

2.4 CMOR

To ensure climate modeling centers can easily produce the netCDF CF metadata convention output, PCMDI has developed the Climate Model Output Rewriter (CMOR) (CMOR, 2009). The structure of the files created by CMOR and the metadata they contain fulfill the requirements of many of the climate community's standard model experiments (Meehl et al., 2007). A newer version of CMOR has been developed for next IPCC model assessment. These features include new grid storage capabilities and additional CF functionalities (Taylor et al., 2008). As a component of CDAT, CMOR has a built-in checker to make sure the model output complies with the netCDF CF conventions.. This checker guarantees the quality of the output of the model data before it is archived into the CMIP5 data repository.

2.5 Climate Data Management System (CDMS)

Implemented as part of CDAT, the Climate Data Management System (CDMS) (Drach et al., 2007) is used to automatically locate and extract metadata (i.e., variables, dimensions, grids, etc.) from the multi-model collection of model runs and analysis files. CDMS is defined as an object-oriented data management system, specialized for organizing multidimensional, gridded data used in climate analysis and simulation. As the only fully compliant netCDF CF data-access tool, CDAT (via CDMS) allows users to seamlessly read data from multiple sources for intercomparison studies. In addition to reading in netCDF CF compliant data, CDAT can also store data in this format. Besides netCDF, CDAT can also read in HDF, GRIB, ASCII, binary, and other popular climate data file formats.

2.6 Data Manipulation

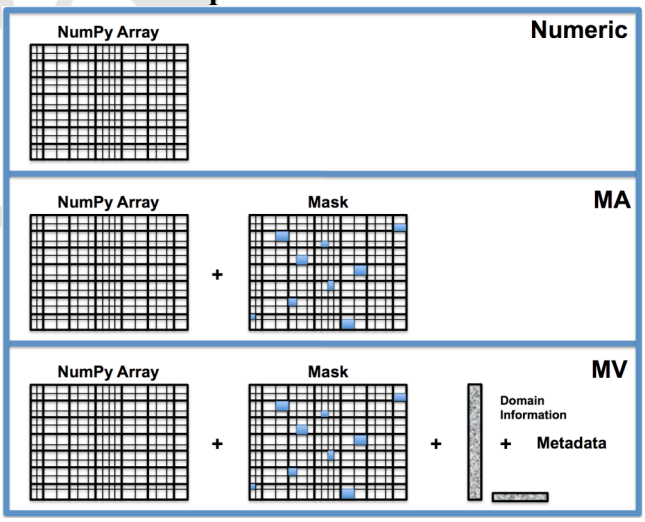

Figure 2: CDAT supports the combination of three array objects: the Python NumPy Array, where all elements in the multidimensional array have the same data type (real, integer, etc.); the Masked Array (MA), a NumPy array with an optional missing data mask; and the Masked Variable (MV) (or Transient Variable), a MA having domain and metadata.

CDAT expedites the manipulation of data arrays with missing values, an often-troublesome issue in geophysical applications. This problem is resolved by extending the Numerical Python ('Numpy') masked-array (MA) function to a masked-variable (MV) construct that retains information on the metadata attributes of data arrays. It also

allows an extensive collection of Numpy mathematical operations to be executed. (See Figure 2.)

2.7 Grids

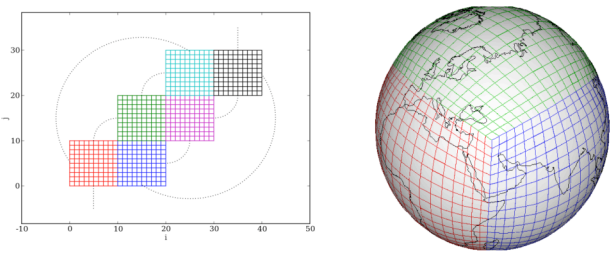

Figure 3: Example of Mosaic Grid: Cubed Sphere Connectivity.

The next generation of multi-model output will not require modeling efforts to supply their data in only latitude-longitude grids as required in previous CMIP archives. Instead, data will be accepted on native unstructured grids. Examples of these Mosaic (or unstructured) grids include the tri-polar, geodesic, and cube-sphere. These new forms of grids will help models overcome the difficulties associated with numerical singularities at the north and south poles. (See Figure 3.)

2.7.1 MoDAVE

To help CDAT deal with unstructured grids, the Tech-X Inc. is developing the Mosaic Data Analysis and Visualization Extension (MoDAVE) to read and interpret Mosaic gridded data (Pletzer et al., 2009). MoDAVE exploits the complex folding patterns between Mosaic gridded tiles to produce quality visualizations. Incorporating MoDAVE into CDAT will allow CDAT to leverage multi-core and graphical processing units to accelerate common post-processing tasks.

2.7.2 SCRIP

The Spherical Coordinate Remapping and Interpolation Package (SCRIP) (Jones, 2001) have been used in CDAT over the years to create remapping and interpolation between any two grids on a sphere. That is, the SCRIP regridder is built into CDAT to interpolate from any latitude-longitude grid (e.g., rectangular or non-rectangular). Originally written in FORTRAN, the new SCRIP software has been rewritten in C and has new capabilities for new grids and new fields. A Python wrapper will be added to the new completed SCRIP package for continued seamless integration into CDAT.

2.7.3 Gridspec

The Geophysical Fluid Dynamics Laboratory (GFDL) gridspec work is also helping to extend CDAT's support of geodesic grids (Balaji, Liang, 2007). The results of the gridspec effort will be implemented in the netCDF CF convention. In addition, gridspec will be implemented in CMOR to improve processing data efforts for CMIP5 model output producing centers. Gridspec is considered to be a standard for the description of grids used in earth system models.

3. DATA ANALYSIS

3.1 Data Analysis Challenges

Scientific research is inherently tied to the researcher's ability to efficiently browse, search, and analyze available data. Recent increases in computation power and storage capacity rendered these tasks especially difficult because researchers are now confronted with huge amounts of data at their disposition. Some of the challenges in analyzing these extreme scale data include: finding out that data exists, accessing data remotely, and processing data for study.

These worldwide-distributed data archives and applications to process and manipulate data pose difficult challenges for the end user. Indeed, once the desired data have been identified, it is not necessarily easy to decide which variables or subsets are required or how to access them with the right tools, let alone determining whether or not the data will be in the right format for analyzing.

There are many tools out there for the researcher to use and no one tool can handle every data situation. As a workbench where many tools are available, the CDAT framework is designed to integrate other tools under one application. In this way, CDAT supports application/module sharing for computation, analysis, and management of large-scale distributed data.

3.2 Contributed Science Committee Packages

3.2.1 NumPy

Originally, CDAT developers came up with their own implementation of the numerical layer. It quickly became apparent that this task was extremely resource-consuming and would be handled significantly better by numerical experts. The developers decided to take advantage of Python interoperability and used Numeric as its underlying numerical core. Nowadays Numeric has been deprecated and has been replaced by NumPy an even more powerful package geared toward very large arrays.

One of the great aspects of Numeric/NumPy is the existence of a sub-package capable of handling "masked" (or missing) data—a problem commonly experienced in the climate community, especially when dealing with observational dataset.

Numpy also includes a f2py—a very powerful tool to incorporate FORTRAN codes to Python.

3.2.2 SciPy

Developed by the same core of people as NumPy, SciPy is an open-source collection of tools for mathematics, science, and engineering, preeminently geared toward signal processing (SciPy, 2009). SciPy is also a community effort and represents a sub-package encompassing: clustering, fast fourier transform, integration and Open Dynamics Engine (ODE), interpolation, input/output (I/O), linear algebra, maximum

entropy models, image processing, orthogonal distance regression, optimization and root finding, signal processing, sparse matrices, sparse linear algebra, spatial algorithm and data structures, statistical functions, image array manipulation and convolution, and C/C++ integration. By default, SciPy is a built-in CDAT module.

3.2.3 RPy

RPy is defined as a very simple, yet robust, Python interface to the R Programming Language (RPy, 2009). It can manage R objects and execute arbitrary R functions (including the graphic functions) from within CDAT. R is an open-source programming language and software environment used for statistical computing and graphics (Girke, 2008). All errors from the R language are converted to Python exceptions and visible in the CDAT environment. Any module installed for the R system can be used from within CDAT. RPy must be specifically requested to be built for CDAT and must point to the correct version of the R libraries.

3.3 Climate Analysis

3.3.1 Genutil

Genutil is a package developed by the PCMDI software developers to facilitate day-to-day climate analysis and diagnosis in CDAT. These tools are not climate specific, but more "array" specific. (See Figure 2.) They are "metadata-smart" retaining metadata information after some sort of data manipulation (a CDAT signature trademark). Example of Genutil tools include statistics, array growing (to expand a data array before comparing data with a different number of dimensions, (e.g. applying a 2D land/sea mask, to a 3D dataset)), color manipulation by name, status bars, string templates, selecting non-contiguous values across a dimension, and other such functions.

3.3.2 Cdutil

Cdutil is another PCMDI developed package that retains metadata information after some sort of data manipulation. It is geared toward climate specific applications such as time extraction, seasonal average, bounds setting, vertical interpolation, variable massager (i.e., prepping variables in order to compare them, such as masking/regridding), region extraction, and other such functions.

3.4 Contributed Climate Packages

3.4.1 PyClimate

PyClimate is a Python package designed to accomplish some usual tasks needed during the analysis of climate variability. It provides functions to handle simple I/O operations, handling of COARDS-compliant netCDF files, Empirical Orthogonal Functions (EOF) analysis, Singular Vector Decomposition (SVD) and Canonical Correlation Analysis (CCA) analysis of coupled data sets, some linear digital filters, and kernel based probability density function estimation (Saenz, Jon, et al., 2004).

3.4.2 General Contributed Package

Bundled in the CDAT distribution is a number of contributed (i.e., "contrib") packages, which are developed at other institutions. To date there are over 30 such packages ranging from I/O to data manipulations to visualizations. These contrib packages are all open-source and can be executed as standalone Python applications or run within CDAT. Some of the more numerically intense packages are the EOF (a powerful FORTRAN subroutine to compute EOF analysis, including weighting for a designated area) and the Microwave Sounding Units (MSU) (a set of tools to produce MSU data from air temperature profiles).

4. DATA VISUALIZATION

4.1 Visualization Challenges

Data visualization is a very important aspect of analyzing climate data and one way by which the user can conceptually comprehend data quickly. With the large-scale data volumes associated with future climate models and other Earth system resources and georeferencing data, it is imperative that utilities exist for easily visualizing and evaluating the data.

Multiple graphics methods and visualization techniques are necessary to aid the climate researcher to comprehend and improve the behavior of global climate simulation models. The corresponding visualization model must be able to access a number of different graphical engines that reside close to the data. (See Figure 1.) In doing so, the user's scope of understanding the climate simulation can be broadened.

4.2 Visualization and Control System (VCS)

The Visualization and Control System (VCS) (Doutraiux et al., 2007) is the *de facto* standard 1D and 2D graphics package for CDAT. It is especially designed for climate change research. Conforming to the netCDF CF convention, VCS allows users have complete control over the graphics. That is, by specification of the desired netCDF CF data, the graphics method, and the display template, the user can control the appearance of the data display, associated text, and animation.

Because it is so flexible, wide ranges of graphical displays are predefined and users can create and share new ones. For the community of MIPs (including CMIP), users have predefined graphical output specific to their diagnostic climate modeling needs.

4.3 Open to Additional Visualization Packages

Because no single type of graphical display is likely to meet all needs, we have elected to seamlessly make available in the CDAT environment several open-source visualization packages. Therefore, besides the longstanding VCS package, users are now able to deploy Xmgrace (Xmgrace, 2008), Matplotlib (Matplotlib, 2009), and IaGraph (Lin, 2004) for 1D and 2D visualizations. 3D

visualizations are allowed via ViSUS (Pascucci et al., 2009), VTK (VTK, 2009) and NcVTK (NcVTK, 2006). (See Figure 1.)

4.3.1 Xmgrace

As an extension of CDAT, the Xmgrace package gives users an alternative choice to producing 1D plots of numerical data. In CDAT, it combines the convenience of a graphical user interface with the power of the Python scripting language which enables it to do sophisticated calculations or perform automated tasks.

4.3.2 Matplotlib

Having origins associated with MATLAB graphics commands, Matplotlib is a 2D graphics library that is freely available in the Python environment. As with CDAT, it makes heavy use of NumPy and other Python extensions,, making it easy to integrate with other CDAT packages. The Matplotlib design philosophy is for users to be able to create simple plots with just a few commands.

4.3.3 IaGraph

Showing the true versatility of Python, IaGraph is a graphics package that is built upon VCS and designed exclusively to make quick plots of atmospheric data. It is designed as a quick interactive graphing tool for displaying line plots, scatter plots, contour plots, and other forms of 1D and 2D graphical displays. IaGraph emulates the commercial Interface Description Language (IDL) (IDL, 2008) syntax.

4.3.4 3D Packages: ViSUS, VTK, NcVTK

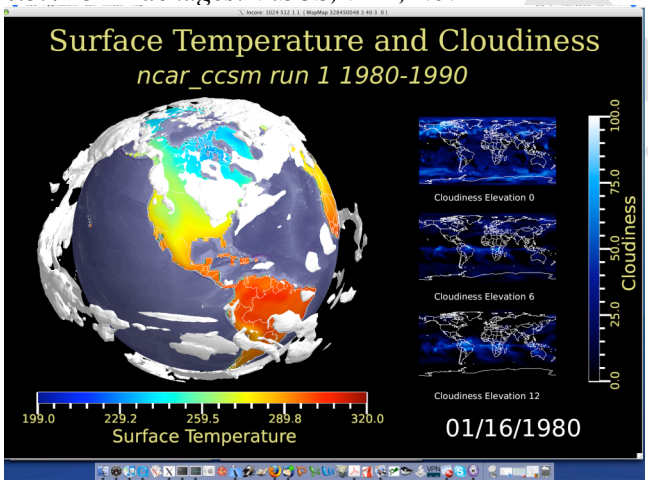

Figure 4: Example of ViSUS 2D and 3D temperature and cloudiness visual output.

Demand has been increasing for 3D plotting capabilities so software packages such as the Visual Toolkit (VTK), the netCDF Visual Toolkit (NcVTK), and the Visualization Streams for Ultimate Scalability (ViSUS) are now being integrated within CDAT. ViSUS (developed in collaboration with the DOE SciDAC-funded Visualization and Analytics Center for Enabling Technologies—VACET) is of particular interest, since its developers are working toward providing future capabilities for petascale dataset streaming. Specialized rendering codes will also be developed to allow simultaneous viewing of multiple 1D, 2D, and 3D representations of high-resolution simulation or observational fields on a single plot. (See Figure 4 and 7.)

5. DIAGNOSTICS

5.1 Diagnostic Challenges

Researchers are uneasy when it comes to sharing their diagnostics. They spend a lot of time putting them together to prove a point, to further the science, or for publication. In order to further prove or disprove climate phenomena, others use these diagnostics. Therefore, eventually bits and pieces of diagnostic codes can be (and are) exchanged with others in collaboration. However, the process of "adopting" a new diagnostic is a very arduous process on both the part of the diagnostic provider and user. Another harmful piecemeal diagnostics consequence is that multiple or similar (but not identical) versions of diagnostics are used, which produce slightly different results. Currently, these diagnostics are being continuously redeveloped to meet a specific I/O format, programming language, or simply because the user doesn't trust external code.

CDAT is trying to close that gap in diagnostic trust and reuse by providing a common environment for all researchers to develop in, and encouraging and facilitating communication and sharing in diagnostic development.

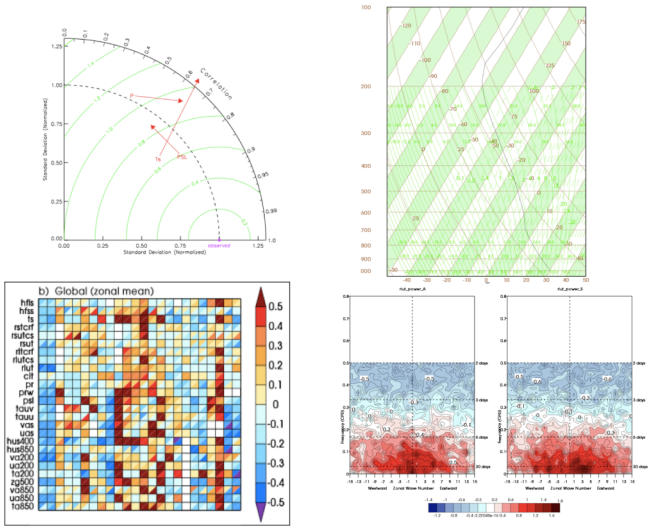

Figure 5: Collage of CDAT diagnostics products. Clockwise from upper-left: Taylor Diagram, Thermodynamic Skew-T plot, Portrait Diagram, and Wheeler-Kalidas plot.

5.2 Taylor Diagram

Developed by Karl E. Taylor, the Taylor Diagram (Taylor, 2001) can provide a concise statistical summary of how well patterns match each other in terms of their correlation, their root-mean-square difference, and the ratio of their variances. (See Figure 5.) Although the form of this

diagram is general, it is especially useful in evaluating complex models, such as those used to study geophysical phenomena. Examples are given showing that the diagram can be used to summarize the relative merits of a collection of different models or to track changes in performance of a model as it is modified.

Methods are suggested for indicating on these diagrams the statistical significance of apparent differences and the degree to which observational uncertainty and unforced internal variability limit the expected agreement between model-simulated and observed behaviors. The geometric relationship between the statistics plotted on the diagram also provides some guidance for devising skill scores that appropriately weigh among the various measures of pattern correspondence.

CDAT is capable of easily producing such a diagram.

5.3 Thermodynamic Plot

The thermodynamic diagram is a tool frequently used by meteorologists to solve atmospheric temperature and humidity problems using simple graphical techniques. (See Figure 5.) Lengthy calculations are avoided since the mathematical relationships have been accounted for in the arrangement of this diagram. Meteorologists use the thermodynamic diagram daily to forecast cloud height and atmospheric stability, the latter of which is an indicator of the probability of severe weather. They base their analyses upon the plots of the vertical profiles of air temperature, humidity and wind that are observed by a radiosonde (a balloon-borne instrument package with a radio transmitter) at individual upper air stations.

CDAT is capable to producing such specialized diagrams, including, but not limited to, skew-T plots, tephigram, or emmagram.

5.4 Performance Portrait Plot

Performance portrait diagrams provide statistical information about test data relative to a reference observational data set. (See Figure 5) In this context, "error" means the difference between the test data and the reference data. The test data can be climate model output, in which case we are measuring model errors (with the usual caveats about the quality of climate observational data). Alternatively the test data can be a set of different or "secondary" observational data. The comparison is presented in a tabular form with the numerical values of the table replaced by shades of colors. The colors indicate the relative size of the rootmean-square (RMS) errors.

5.5 Wheeler-Kalidas Analysis

Based on the Wheeler-Kalidas paper, this diagnostics performs and space-time fast Fourier transform (FFT) analysis, and also produces the adequate figures to represent the results. (See Figure 5.)

6. SCRIPTING AND USER INTERFACE

6.1 Scripting and User Interface Challenges

Scientists will always to have pressing needs to write programs to ingest, manipulate, and display data—and repeat these very same tasks as new data becomes available. Therefore the task at hand is to minimize this effort as much as possible so that scientists can focus on research. To this end, CDAT makes use of an open-source, object-oriented, easy-to-learn scripting language (Python) to link together separate software subsystems and packages to form an integrated environment for climate data analysis. This resulting software environment is user-friendly, reusable, portable, and promotes the sharing of common standards within the community. In this environment, specialized data access, diagnostic algorithms, and display software cooperate under a variety of user interfaces including command line, standalone application scripts, graphical user interfaces, and web browsers.

6.2 Python Programming and Scripting

Besides being able to enter interactive command line calls at the Python prompt, users can produce a complete Python script (or series of commands) to a file and save it. This script can be executed at anytime, reopened and modified, and/or shared with colleagues. Scripts save the user from doing repetitive actives by describing exactly which commands to execute and in what sequence, so that the operations on the data are doing the right thing and in the correct sequence. Python is widely used in other scientific application areas and there are many books, tutorials, and websites on this popular fourth generation language. To learn more about the Python programming language, check out the two books titled, “Core Python Programming” and “Python Phasebook”. Also check out the Python website at: http://www.python.org.

6.3 Visual Climate Data Analysis Tools (VCDAT)

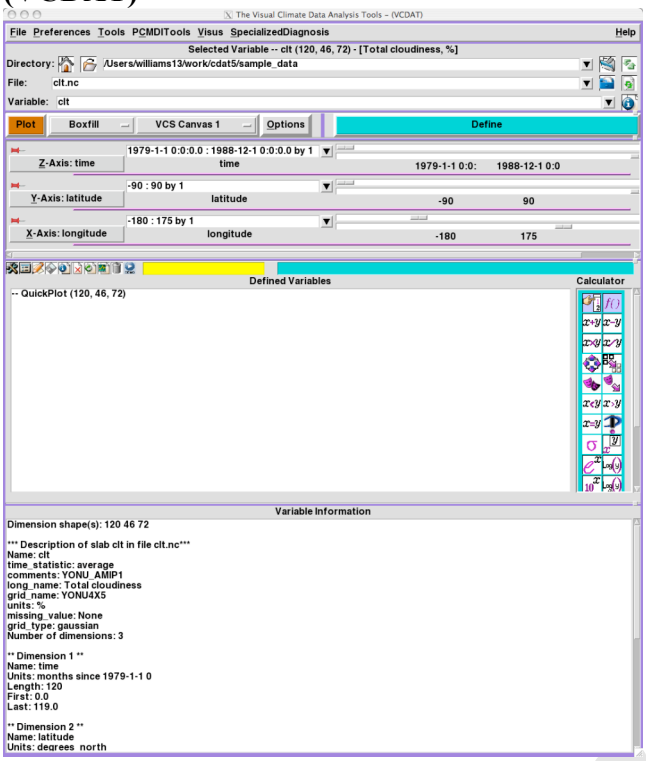

Figure 6: A view of the current Visual Climate Data Analysis Tools (VCDAT) user interface.

The graphical user interface (GUI) for CDAT is called the Visual Climate Data Analysis Tools (or VCDAT). (See Figure 6.) It can be used for quickly accessing and computing data, producing a picture that visually represents the data values, refining the plot, and saving the state of a session so that it can be reused later. In addition, VCDAT comes with online help and allows the user to enter command line instructions and write and execute stand-alone application scripts. It is important to note that VCDAT does not require learning Python or the CDAT software. Eventually, when users have learned to use VCDAT and need to extend its capabilities and write their own scripted applications, VCDAT helps users become familiar with CDAT by translating nearly every button press and keystroke into Python scripts. All scripts include comment lines to assist the user. The script file can be saved and later run and modified by the user, thus helping the user learn how to write customized CDAT scripts. This facility also allows the non-interactive repetition of common tasks.

6.4 VisTrails

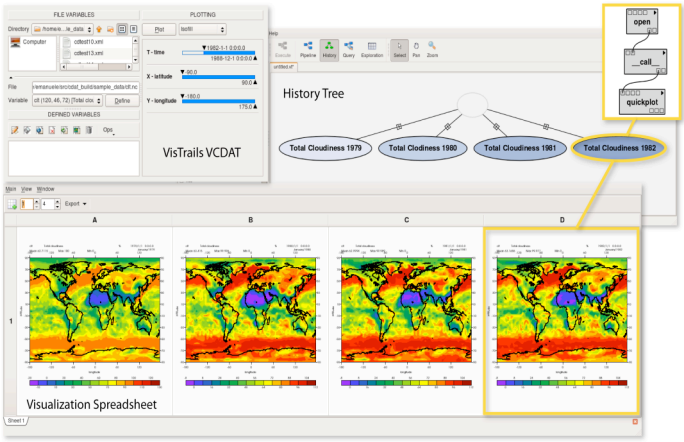

Figure 7: Example of CDAT workflow built inside of VisTrails.

Developed at the University of Utah, the scientific workflow and provenance management system called VisTrails (Santos et al., 2009) is merging with CDAT. (See Figure 7.) The resulting CDAT VisTrail package will give users the capability to perform comparative visualizations through the CDAT infrastructure while maintaining provenance (i.e., source and ownership history) information during the exploration and visualization of data. In this collaborative merge, the users will see the familiar VCDAT interface along with additional tools to build their workflows and to automatically capture provenance. In addition, the user will see VisTrails history trees, data pipelines, and an impressive visualization spreadsheet for displaying 1D, 2D, and 3D interactive graphics. This featured functionality that supports concurrent exploration of multiple visualizations with the use of a spreadsheet allows users to effectively compare visualizations produced by different workflows side-by-side. Indeed, the integration of CDAT, ESG, and VisTrails facilitates a part of an overall solution that manages and analyzes large and diverse datasets worldwide.

6.5 LAS

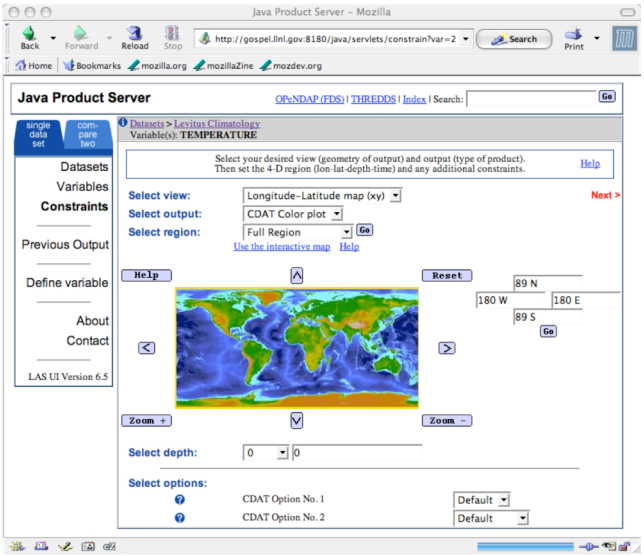

Figure 8: Example of the LAS selecting CMIP3 (IPCC AR4) data. CDAT product options are returned as a result of the selection.

A prime example of CDAT's external collaboration is the integration of the Live Access Server (LAS) to facilitate exploration of geophysical datasets through a user-customized web browser. In this integration, CDAT supplies the necessary server-side processing to produce advanced diagnostic output. Because LAS is integrated into the ESG data infrastructure and CDAT is integrated into LAS, ESG-based CMIP5 data manipulation will be accomplished via CDAT commands (Mlaker et al., 2009).

7. CMIP5 DATA

Modeling centers are scheduled to start producing CMIP5 model output by January 2010, with the final production of data ending in early 2011. The output will consist of three suites of experiments covering "Near-Term" decadal predictions, "Long-Term" century and longer predictions, and "Atmosphere-Only" runs. From the three suites of experiments, 17 modeling groups have registered over 40 different model variations. In total, these models are projected to produce between 6 to 10 petabytes of uncompressed data. Of this amount, 600 terabytes to 1 petabytes of core data will be stored centrally at PCMDI for IPCC analyzes study.

7.1 ESG Data Node and CDAT

As modeling centers start their generation of CMIP5 data, PCMDI will require data providers to place their data on rotating disks at their site for fast access. An ESG "Data Node" software stack, consisting of several software components including CDAT, will be provided for data processing and harvesting. The "ESG Data Node" is a host on which a data archive resides, or from which an archive is published. If desired, the data provider can install the ESG product server containing CDAT to allow users server-side analysis capabilities.

7.2 ESG Data Publisher

Once the ESG software stack is in place, the data provider will run the ESG Data Publisher to catalog and make data visible for search, download, and analysis from the PCMDI ESG "Gateway". Before publishing data, CDAT's built-in CMOR checker will be used to test the quality of the data output and to make sure it is netCDF CF compliant. CDAT components are also used in the data publisher to scan and catalog the data for its physical location. The PCMDI ESG "Gateway" is the community facing web presence that handles user registration, discovery, and brokers data requests and other resources.

8. CONCLUSION AND FUTURE WORK

CDAT's most recent notable accomplishment has been in the production and processing of the multi-model climate simulation data used by scores of scientists who contributed to the recent IPCC AR4. Other national and international MIPs including the Atmospheric MIP, the Coupled MIP, the Seasonal Prediction MIP, the Cloud Feedback MIP, and the Paleoclimate MIP, have used CDAT for data production, processing, and analysis. In the coming years, CDAT will lead the way in processing and analyzing data for the CMIP5 for scientists contributing to the forthcoming IPCC AR5.

9. REFERENCES

1. Balaji, V., Liang, Zhi, 2007: Gridspec: A standard for the description of grids used in Earth System models, http://www.gfdl.noaa.gov/~vb/gridstd/gridstd.html.
2. CF, Climate and Forecast (CF) metadata convention for processing and sharing data files (2009). http://cf-pcmdi.llnl.gov/.
3. CMOR, the Climate Model Output Rewriter produces CF-compliant netCDF data files (2009). http://www2-pcmdi.llnl.gov/cmor.
4. Doutriaux, Charles, Drach, Robert, Williams, Dean, 2007: The Visualization and Control System (VCS), http://www2-pcmdi.llnl.gov/cdat/first-page/cdoutrix/beginners-guide/data-visualization/?searchterm=VCS.
5. Drach, Robert, Dubois, Paul, and Williams, Dean, 2007: Climate Data Management System, version 5.0, http://www2-pcmdi.llnl.gov/cdat/manuals/cdms5.pdf.
6. Gates, W.L., J.S. Boyle, C. Covey, C.G. Dease, C.M. Doutriaux, R.S. Drach, M. Fiorino, P.J. Gleckler, J.J. Hnilo, S.M. Marlais, T.J. Phillips, G.L. Potter, B.D. Santer, K.R. Sperber, K.E. Taylor, and D.N. Williams, 1999: An Overview of the Results of the Atmospheric Model Intercomparison Project (AMIP I). Bull. Amer. Meteor. Soc., 80, 29–55.
7. Girke, Thomas, 2008: Programming in R, http://www.faculty.ucr.edu/~tgirke/Documents/R_BioCond/R_Programming.html.

8. HDF Group, HDF: a file format used for storing various forms of data (e.g., raster, scientific, etc.) (2009). http://www.hdfgroup.org/.
9. IDL, 2008, http://www.ittvis.com/ProductServices/IDL.aspx.
10. Lin, Johnny, IaGraph, 2004, http://www.johnny-lin.com/py_pkgs/IaGraph/Doc/index.html.
11. Lutz, Mark, "Learning Python: Powerful Object-Oriented Programming", Third Edition (2007). Published by O'Reilly & Associates, Copyright.
12. Matplotlib, 2009, http://matplotlib.sourceforge.net/.
13. **Meehl, G.A., C. Covey, T. Delworth, M. Latif, B. McAvaney, J.F.B. Mitchell, R.J. Stouffer, and K.E. Taylor, 2007: THE WCRP CMIP3 Multimodel Dataset: A New Era in Climate Change Research. Bull. Amer. Meteor. Soc., 88, 1383–1394.**
14. Jones, Philip W., 2001: SCRIP A Spherical Coordinate Remapping and Interpolation Package, version 1.4, http://climate.lanl.gov/Software/SCRIP/SCRIPusers.pdf.
15. NcVTK, 2006, http://ncvtk.sourceforge.net/.
16. Pascucci, Valerio, Lamar, Eric, Laney, Daniel, Lindstrom, Peter, Duchaneau, Mark, Frank, Randall, Scorselli, Giorgio, Bremer, Dave, Bremer Peer-Timo, 2009: Visualization Streams for Ultimate Scalability (ViSUS), http://www.pascucci.org/visus/.
17. Pletzer, A., Hamill, P., Muszala, S., Williams, D., Doutriaux, C., 2009: MoDAVE: Analyzing and visualizing Mosaic climate data, http://esg-pcmdi.llnl.gov/review-folder/collaborations_partnerships/ESG-MoDAVE.pdf.
18. RPy, 2009, http://rpy.sourceforge.net/.
19. Saenz, Jon, Fernandez, Jesus, Zubillaga, Juan, PyClimate: 2004, http://starship.python.net/crew/jsaenz/pyclimate-1.2.1/.
20. E. Santos, H. Vo, T. Fogal, D. Williams, C. Doutriaux, and C. Silva, Streamlining Data Exploration and Visualization in Climate Science, 2009, http://esg-pcmdi.llnl.gov/review-folder/collaborations_partnerships/VisTrails-CDAT-ESG-2-pager.pdf.
21. SciPy, 2009, http://www.scipy.org/.
22. Taylor, K. E. (2001), Summarizing multiple aspects of model performance in a single diagram, J. Geophys. Res., 106(D7), 7183–7192
23. Taylor, K.E., R.J. Stouffer, and G.A. Meehl, 2008: Summary of the CMIP5 Experiment Design, https://pcmdi-cmip.llnl.gov/cmip5/docs/Taylor_CMIP5_design.pdf.
24. UCAR, University Corporation for Atmospheric Research, netCDF: a machine-independent, self-describing, binary data format (2008). http://www.unidata.ucar.edu/software/netcdf.
25. Velimir Mlaker, Dean N. Williams, Roland Schweitzer, and Steve Hankin, Remote Analysis of Climate Model Output, 2009, http://esg-pcmdi.llnl.gov/review-folder/collaborations_partnerships/ESG-LAS-CDAT_v3.pdf.
26. VTK, 2009, http://www.vtk.org/.
27. D N Williams, R Ananthakrishnan, D E Bernholdt, S Bharathi, D Brown, M Chen, A L Chervenak, L Cinquini, R Drach, I T Foster, P Fox, D Fraser, J Garcia, S Hankin, P Jones, D E Middleton, J Schwidder, R Schweitzer, R Schuler, A Shoshani, F Siebenlist, A Sim, W G Strand, M Su, N. Wilhelmi, "The Earth System Grid: Enabling Access to Multi-Model Climate Simulation Data", in the Bulletin of the American Meteorological Society, February 2009.
28. Williams, Dean N. (1997). The PCMDI Software System: Status and Future Plans of CDAT (Climate Data Analysis Tools), PCMDI Report No. 44. http://www2-pcmdi.llnl.gov/cdat.
29. WMO, World Meteorological Organization standardized the GRIdded Binary (GRIB) data format commonly used in meterology to store historical and forecast weather data (2009). http://www.grib.us/.
30. XmGrace, 2008, http://plasma-gate.weizmann.ac.il/Grace/.