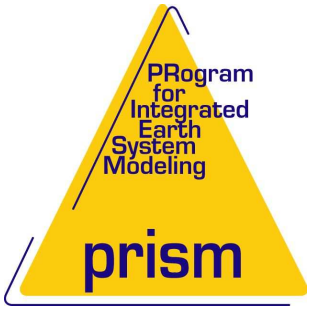
PRISM

Project for Integrated Earth System Modelling

An Infrastructure Project for Climate Research in Europe

funded by the European Commission

under Contract EVR1-CT2001-40012



Future Development of PRISM Visualisation tools

Edited by:

Steve Mullerworth and the Processing and Visualisation Team

PRISM-Report Series-16

1st Edition

(last change: December 17, 2004)

How to get assistance?

The individual work packages of the PRISM project can be contacted as listed below.
PRISM publications can be download from the WWW server of the PRISM project under the URL: <http://prism.enes.org/Results/Documents/>

Phone Numbers and Electronic Mail Adresses

Electronic mail adresses of the individual work packages are composed as follows :
prism_ *work package* @prism.enes.org

Name	**Phone**	***PRISM Work Package***
Steve Mullerworth	+44-1392-886899	*wp4a*
The Processing and Visualisation Team		*wp4a*

Contents

List of Figures

List of Tables

Chapter 1

Future Development of PRISM Visualisation tools

1.1 Introduction

Tools and techniques for processing and visualizing climate model data have been reviewed, extended and developed to prove that they can meet the wide range of requirements of climate modellers. This document tries to point the way forward in how these tools need to be adopted and adapted to meet the current and future requirements of climate scientists.

- Development of the coco tool for processing data in the netCDF(CF) format which was the chosen PRISM format.
- Use of the OpenDX and VTK graphics packages to do interactive visualization of PRISM data.
- Methods for archiving data and for accessing archived data.
- Systems through which a user can select plots to be automatically generated while a model is running.

The demonstration programs that have been developed during the PRISM project prove that the areas of work have the potential to be developed into effective tools. But the diversity of requirements that exist among the full spectrum of climate research means that work is needed to enable the chosen packages to cope with the diverse grid types, plot types, processing requirements etc. that are needed.

It is not essential for a climate research group to adopt the whole set of tools evaluated during PRISM as, depending on requirements, each area can be considered independently. The area that unites much of the work is the choice of PRISM data format of netCDF(CF) that all the tools must be able to read, and the support of the Python scripting language of some of the tools.

All of the tools, and pointers to further information about them, have been documented in the Data Processing and Visualization chapter of the main PRISM technical document. The following sections describe ways in which each of the areas may be adopted or adapted, and any essential work that needs to be done.

1.2 coco

Development of the coco processing tool is likely to continue at the Met Office beyond the end of the PRISM project, as the Met Office currently intend to adopt it in the future. Contact the Met Office for further information.

Some immediate development of coco is required to make it fully compliant with the latest version of CDAT that was released too late in the project to complete.

In terms of further development, coco was originally developed using CDAT as a basis. However, CDAT has been further develop since the start of the project and there are continuing opportunities to incorporate the aims of coco directly into CDAT, which would reduce the effort required in maintaining the software.

The coco package is written in Python and C++ and has a Python interface. While Python is an object-oriented language, this should not be a concern for users of coco who should find the Python interface easy to pickup. However, experience developing coco suggest that highly qualified staff, with good experience in object-oriented programming are required for developing coco further, and development time should not be under-estimated. People wishing to develop coco further should remember that some people take more time to understand how to use object-oriented languages effectively.

1.3 OpenDX and VTK

Both OpenDX and VTK are mature graphics package with a wide user community outside of the climate community. Much of the work within PRISM has concentrated on ensuring these tools can read and understand the PRISM data format and on demonstrating their abilities with example applications. This work has been described within the main PRISM document and some ideas for specific improvements have been made.

The differences between the two packages have been described elsewhere in the PRISM documentation. Much of the focus on OpenDX is its Visual Program Editor that helps build visualisation tools based on user interfaces and direct interaction with the graphics. VTK supports the Python interface enabling it to be used within a scripting interfaces which is particularly useful when supporting automatic generation of plots, though the VTK examples have shown that it can be used to good effect with the Qt application development framework to build an effective graphical user interface based application. Different institutions have differing requirements and preferences regarding such aspects of tools like these.

To take either of these packages forward, support structures are essential to provide effort to maintain the PRISM-specific software, and to develop and support new software to meet any new user requirements. Maintenance is required to ensure the PRISM specific software modules, packages and documentation remain up to date with changes to the core graphics packages themselves. For example, the demonstration applications for VTK additionally incorporate other existing software packages such as Qt. As new versions of packages emerge, the updating of existing applications has to be considered. Development of new software or new functionality is required because, as is typical with data visualization, each community of users will have different needs.

VTK lends itself for close integration with coco and CDAT and care should be taken to ensure this is done. To date, integration of coco and CDAT with OpenDX is limited to a common file format (data process by coco can be read by OpenDX). Opportunities for further integration should be explored, such as sharing IO software and direct data communication through the Python binding available in OpenDX (Py-Opendx).

1.4 Archiving Data

Through use of tools such as OPenDAP, described in the next section, institutions can maintain their own archives of data in the format most suitable for themselves while also setting up a server to allow access to external users to all or parts of their database.

In setting up new data archive systems, there are opportunities to agree on a suitable format for archived data. Many tape based archives have volume based license charges and so a format that allows data to be stored using lossy or non-lossy compression is important. Further, because data is accessed across a number of dimensions it is important that the atom of retrieval is defined at a low enough level to avoid the need to restore large amounts of data when only a small proportion of it is required. Again, agreeing

on standards and technologies that can be used across a number of proprietary solutions is an important, but non-trivial task.

Projects that are examining the issue of providing distributed networks of data archives include the DataGrid project which was funded by the European Union and came to an end in 2004. This project looked at ways of enabling access to geographically distributed computing power and storage facilities belonging to different institutions, and considered applications for Earth observation among others. See the following link for further details:

`http://eu-datagrid.web.cern.ch/eu-datagrid/`

1.5 OPeNDAP

Use of the OPeNDAP system to set up an archive of PRISM data which can be accessed using internet protocols has been demonstrated. OPeNDAP was originally known as DODS, the Distributed Oceanographic Data System, and renamed because it has more general applications than its original use. The intention of OPeNDAP is to encourage climate data to be stored where it is generated, but for it to be easily accessed by remote users. This policy results in the creation of highly distributed systems where each institution within a research community may host a separate database. By ensuring each database has an agreed and simple protocol for accessing data OPeNDAP makes it easier for user applications to be modified to access data on OPeNDAP servers. This means that users can use their application of choice to access and analyse data in any OPeNDAP archive.

The OPeNDAP philosophy would suggest that each institution running PRISM climate model simulations installs its own OPeNDAP system and makes data available to external users through the system. The OPeNDAP system is flexible enough to allow it to be adapted to differing data storage policies, technologies and interfaces, that are likely to exist at different sites. The differences in each system will not affect how data is accessed through the web interface. The OPeNDAP philosophy would also encourage partners to modify their own internal data analysis software to allow simple access to data in external archives. Clearly any applications developed around PRISM visualisation tools should be designed with this facility; the ability to incorporate this functionality was shown by the PRISM demonstration applications such as VTK_Mapper.

Climate models produce very large amounts of data that cannot be stored on line and require tape based storage systems. It is suggested that when setting up online access, consideration is given to advising people requesting data of the estimated access and download times they are likely to experience.

1.6 Low End Visualization

The PRISM low end demonstration application showed a basic way of automatically generating a list of plots as selected by a user through the PRISM user interface. The list of available plots was defined by the options in the user interface and the functionality of the underlying visualization script. The options available within the system can be enhanced by adding additional functions to the existing script or by writing new scripts. These new options need to be reflected in the PRISM user interface. The ParaGen system should be used to convert the script interface definitions into a PRISM user interface window in an appropriate way.

Since users do not have direct access to the low end visualisation servers, they have less control over the options for generating plots. Therefore, it is recommended that effort should focus on supporting the general requirements of a wide range of users rather than specific requirements of one or two. This can be done by concentrating on the need to provide simple plots of a wide range of model products rather than attempting to generate plots for a small number of model products that require a lot of complex processing.

Users will need to have their requirements considered when specifying what options for low end visualization are available. Procedures need to be in place to allow users to submit new visualization scripts for possible installation on the visualization server.

The current example low end systems generated plots from models as they were generated. Other systems can be envisaged where data may be archived before it can be accessed or where low end systems are set up to access data in archives rather than from current model runs. In consideration of these aspects, work in developing the Low End interface should be coordinated with work on OPeNDAP systems and data archive systems.