

Steps Towards a Common Data Exchange Infrastructure in the Bluelink Project

Tennessee Leeuwenburg, Bureau of Meteorology

Keywords: BLUElink, oceanography, data management system

Correspondence: postal: Mr Tennessee J Leeuwenburg, National Meteorological and

Oceanographic Centre, PO Box 1289K, Melbourne, Victoria 3001. email:

t.leeuwenburg@bom.gov.au phone: (03) 9669 4310 fax: (03) 9669 4460

Abstract

The Bluelink Project aims to provide improvements to ocean forecasting in Australia. With new infrastructure technologies being implemented, as well as the upgrade of data collection mechanisms and science components, there are implications for data management issues across the board.

The scientific improvements have involved the coupling of ocean and atmospheric modeling in order to capture the ocean-atmosphere interaction more fully, providing higher quality model predictions, providing data to the models at the best resolution achievable by the observation network, and sharing the results. This has required the integration of many previously separate systems.

Equal efforts by the CSIRO and the Royal Australian Navy (RAN) have involved a high level of cross-institutional involvement, which has meant that standards for data exchange have had to be

put in place. Data management involves many different data formats, with a matrix of conventions and compatibilities, which can mean that choosing a single data representation and exchange mechanism may be impossible. Within Bluelink, we have adopted a technology called OPeNDAP¹ for exchanging data between institutions in a fast and efficient way, and gained a greater understanding of the variety of scientific data standards along the way. As well as data sharing between institutions, ancillary applications for data presentation also have their own compatibility issues and challenges.

It is very nearly possible to achieve a ubiquitous exchange mechanism using OPeNDAP servers, open software, and various data conversion utilities. OPeNDAP can function as an insulator, allowing unusual back-end solutions to interface successfully with the wider world.

Table of Contents

Steps Towards a Common Data Exchange Infrastructure in the Bluelink Project.....	1
Abstract.....	1
The Problem at Hand.....	3
Existing Systems.....	4
Data Transport Considerations.....	4
Data Format and Structure Considerations.....	5
Implications for New Projects.....	5
The OPeNDAP Technology.....	6
The Potential of Distributed OPeNDAP Networks.....	7
Reducing Network Overhead (Data Transport).....	7
Data Encapsulation (Data Format and Structure).....	9
OPeNDAP Client-Side.....	9
Integration of OPeNDAP servers with data applications – Live Access Server (LAS).....	10
Integration of OPeNDAP servers with data applications – Unidata IDV.....	11
Integration of OPeNDAP services with other data applications.....	12
Prototype System Design And Assessment.....	12
Building the Prototype.....	13
...Data storage.....	13

¹ <http://OPeNDAP.org/>

...Data formatting.....	14
...Sharing and Publishing Data.....	14
...The OPeNDAP Server.....	15
...Securing the System.....	16
Observed Benefits.....	17
Observed Problems.....	17
Expectations / Success Criteria.....	18
Going Forward.....	18
...The OPeNDAP Community.....	18
...Best Practises (Implementation Suggestions).....	19
Closing Remarks.....	20

The Problem at Hand

The wide variety of systems which needed to be integrated in the Bluelink project meant that significant work needed to be done in order to draw from all available data sources. In no particular order, some of these sources are: satellite data; ship-based data; floating buoy data; tide gauge data; atmospheric model coupling; historical data sets. Software using several sources needed to write a specific interface to each.

The independent development of many of these systems in the past has led to different data formats being adopted in each. This wide variety suggested that finding a unified interface could simplify the task. A framework for understanding interoperability problems was developed by Cornillon et al², under the title “Layers of Interoperability.”

Metadata – Map to consistent metadata
Data Modification – e.g. Unit conversion
Semantic Structure – more consistency i.e. conventions
Structure – Map data to a consistent structure
Format – Map data from source to application
TCP, IP Layers
Hardware Layers

2. OPeNDAP: Accessing data..., *Data Science Journal*, Volume 2, October 2003

This framework separates out interoperability into distinct areas of data management. TCP and IP have been the standard means for networked data exchange, but problems of format, structure, semantic structure, data modification and metadata continue. Before describing the ways in which OPeNDAP allows us to substantially resolve some of these problems, it is first worth considering existing systems in the light of this framework.

Existing Systems

Data Transport Considerations

Based on Bureau usage statistics, the most widely used communications protocols are HTTP followed by FTP. Based on knowledge of the systems, the HTTP usage is likely to serve people who are looking for weather products – such as weather forecasts, radar images, warnings etc – while FTP usage is more likely to be used for the transfer of weather data – such as numerical model output, analysis data, direct observations etc. During the year 2004/5, 23 Terabytes of HTTP data are uploaded from the Bureau, compared with 5.4 Terabytes of FTP data.

In addition to the issues surrounding the efficiency of transport, there are also attempts to cut down the problem of transporting more with each file than one needs. Common practise is to break very large data sets into subsets based on operational requirements. Each new data need then necessitates the publication of a data subset designed specifically to meet those needs.

Nonetheless, these methods address the problem of data transport. Issues of parsing and interpreting the transferred data remain. In this area, software continues to be written handling specific data sources rather than generalised data per se. Until data formats can be rationalised, this problem will continue.

Data Format and Structure Considerations

Managing disparate data sources is not a new problem, so it is perhaps surprising that it form a large part of any particular project. To understand why such a unified interface as hinted at earlier is not already commonplace, it is instructive to consider the “geography” of meteorological data.

Historically, decisions about data formats were made chiefly with regard to communications efficiency. The cost of transmitting the large volumes of data led to instrument-specific binary message encodings. Having a format that was highly compressed for a known message format was simply necessary.

This led to a balkanised group of message formats, peculiar to specific instruments and purposes. Parsing and interpreting these message formats was a task then pushed onto the collecting institution. Message decoders were written for specific tasks, often as incomplete solutions which addressed only the task at hand.

As operational systems grew out of these software, additional formats would be added as needed.

While some standardisation has occurred, the data format space is still largely divided according to purpose and message type.

Implications for New Projects

There are two implications of this. Firstly, new projects crossing data format boundaries will involve a certain amount of effort to aggregate heterogeneous sources.

The second is that for each user need, existing technologies largely necessitate the creation of products specific to those needs. The data supplier has to do additional work to meet each new kind of client request.

Simple protocols like FTP, HTTP etc do not address format or semantic needs. Fortunately, a new technology was identified which purported to solve some of these problems entirely.

The OPeNDAP Technology

OPeNDAP is, essentially, a protocol for exchanging scientific data over HTTP. In the context of the interoperability framework it addresses format and structure issues, although this still leaves the burden of semantic-structure and metadata. Moreover, it improves data transport efficiency by enabling server-side operations on the data prior to transfer. This allows a client to request not merely the files they need, but the data they actually want. This gives providers a flexibility and efficiency which is unattainable through file-transfer mechanisms alone.

The relevant advantages of using an OPeNDAP server are :

- Accepts many data types for resource data
- Presents heterogeneous data sources in an aggregated way
- Permits data-volume savings through server-side data operations
- Provides flexibility to the client by allowing multiple views of the data
- Supports client libraries in many languages

A network of OPeNDAP servers can in theory serve as both incoming and outgoing proxies, each allowing the aggregation of both local and remote data sources into a unified view of available data.

However, implementing an OPeNDAP solution is not a matter of simply unwrapping the box.

There are a number of challenges, both technological and social. To show why it is worth meeting these challenges, let us first attempt to quantify the value of doing so.

The Potential of Distributed OPeNDAP Networks

The benefit of adopting an OPeNDAP solution is largely in the improvements exhibited over alternative data distribution models, especially for the identified use cases below.

Reducing Network Overhead (Data Transport)

A problem faced by all institutions wishing to exchange a large data volume is that of cost. While the Internet has brought with it new levels of access to and speed of information retrieval, very high data volumes remain expensive to move. Both the dollar cost of the service, and the performance costs of network congestion must still be born.

As alluded to earlier, OPeNDAP servers enable cost savings, which can be maximised when particular network configurations are used.

The greatest cost savings are found where the amount of data available is high, but the breadth of data requested is low. Server-side subsetting of data means that only the data required by the end user need be transferred. Where all data must be replicated across the entire network, OPeNDAP is little different to traditional distribution mechanisms such as automated FTP transfer. If however full replication is not required, then OPeNDAP allows significant reductions of volume.

Let us take an example with 5 institutions, each with 5 researchers requiring access to data X, which happens to be 5GB in size. The resulting network will be a star-network with a single institution being the primary data source, and four other institutes replicating some of that data for their own uses. The researchers desire a subset of data from their host institutions. Common practise breaks up major data sets into multiple files, perhaps by level, variable (or set of variables) etc. This allows them to request only the files which contains the information they ultimately want. An example of

this would be regularly downloading the data for an entire forecast model, but not downloading other, unrelated data sets. This is more efficient than fully replicating the entire source database, but may still represent significant unnecessary duplication. In this instance, let us assume the cost is 4 * 5GB for the initial replication to each requesting institution while internal transfers are free.

Imagine how this same example might look if a network of OPeNDAP servers were put in place instead. Instead of replicating data between all 5 institutions, only the meta-data, or catalog, need be copied, at trivial cost. An institutional researcher's typical need might be for a single level of pressure data, high resolution data over an area of interest, or low-resolution data across a wide area. Let us assume that the requirement in volume terms is for 80MB of data. Instead of needing to transfer 5GB each to meet that potential need, the four requesting institutions need only the actual 80MB of data required. Instead of 20GB net traffic, instead we have $80\text{MB} * 4 = 320\text{MB}$, a reduction to just 1.6%. The time taken to transfer the data is similarly reduced.

If the usage pattern is genuinely of this kind, and institutions do not require a complete archive of the data, then massive savings may be made. It is currently unknown how much of existing data exchange could take advantage of the new systems. It is certainly under-utilised in practise, even where OPeNDAP is an accepted protocol. For example, the OPeNDAP server at NOMADS³ has served out 1.4 Terabytes during the 2004/5 calendar year. During this time, just 15 requests were made for data subsets. This shows a number of things : that OPeNDAP scales to large data volumes, that it being taken up by the community and that its benefits are still not being fully utilised.

While there will always be a need for full replication of some data sets for operational purposes, this is not always true for inspection and research purposes. Using OPeNDAP technology makes it cheaper for the host institution to provide data, which goes a long way to justifying any initial

³ <http://nomads.ncdc.noaa.gov:9090/dods>

implementation costs.

Data Encapsulation (Data Format and Structure)

Adopting an OPeNDAP solution allows the possibility of reducing workload when integrating a new data source, or interacting with a new institution. Using a common standard allows all parties to reduce the cost of interaction.

Unlike the mere adoption of a common data format, OPeNDAP technology permits each institution to work using their favourite format or a basket of formats internally, but still maintain the goals of low-cost interaction with other institutions and ease of use.

The OPeNDAP format is highly flexible, meaning that there are few insurmountable difficulties in representing information. Existing server implementations already support a variety of back-end data sources, but not an exhaustive number.

The decision to implement an OPeNDAP solution is a matter of cost of any customisation and initial set-up versus the savings gained by presenting a consistent front to external client.

OPeNDAP Client-Side

OPeNDAP serves out several kinds of data – html, xml, ascii and dods. Visiting an OPeNDAP server with a web browser will give you a web page interface for requesting data, which can then be downloaded to local disk for later use, or simply to find the links that your application requires.

This allows a kind of minimal level of data discovery. However, the nature of data publishing is such that people usually look for application integration. This is where the XML features of

OPeNDAP come into play. While it does not offer an XML RPC style web service, the catalog is published according to a standard, and as such supports automated data discovery.

Two common client applications are the Live Access Server (LAS)⁴ and the Unidata Integrated Data Viewer (IDV)⁵.

Integration of OPeNDAP servers with data applications – Live Access Server (LAS)

LAS is a both a viewing application and a retrieval tool, implemented using cgi scripting and javascript. The result is an interactive map, showing you a thumbnail image of the rendered data.

From the developer's page page: *“The Live Access Server (LAS) is a highly configurable Web server designed to provide flexible access to geo-referenced scientific data. It can present distributed data sets as a unified virtual data base through the use of [DODS networking](#). Ferret is the default visualization application used by LAS, though other applications (Matlab, IDL, GrADS, ...) can also be used.”*⁶

One of the major strengths of LAS and OPeNDAP is that once installed, publication of additional products becomes a relatively painless exercise. The setup of additional data sets is no longer a complex matter of setting up FTP servers, information pages, user security etc. Certainly security mechanisms exist, but they are a part of standard packages. Using an apache/tomcat/ldap rollout, setting up additional users and groups fits easily into existing organisational policy. If a less organised approach to security is required, simple directory-level password protection is also possible.

4 http://ferret.pmel.noaa.gov/Ferret/LAS/ferret_LAS.html

5 <http://my.unidata.ucar.edu/content/software/IDV/index.html>

6 DODS is the historical name for the now OPeNDAP project. It is an acronym for “Dynamic Oceanographic Data Server”, and there are still many references to this title in use.

Indeed, there is an extent to which the entire process from publication of data, to replication and discovery, can be entirely automated. Public access LAS servers at NVOODS⁷ and NASA⁸ demonstrate the principle. The technology scales from casual access to large-scale data sharing needs. LAS provides access to data in standard scientific formats such as NetCDF, views of data for visualisation purposes, and visual subsetting capabilities.

Integration of OPeNDAP servers with data applications – Unidata IDV

The IDV is a client visualisation application, written in Java. It is a tool capable of rendering a diverse range of data structures in 5 dimensions (spatial, animation and colour). It supports data discovery via OPeNDAP catalogs at run-time.

The IDV demonstrates that using an OPeNDAP server can transport data from the source directly into application. While this functionality does depend on certain conventions being followed in the structure of the source data, it does provide a way to abstract ever more of the underlying technology involved with not just transferring data, but making use of it.

Illustration 1 shows the IDV presenting a 3D view of air temperature. The plot shows the barometric height of a particular temperature (hence the fairly uniform colouring), with variation in the Z axis showing the height. The data was able to be imported into the application without having to specifically manage either the transport layer or data format.

7 [Link to NVOODS LAS server](#)

8 [Link to NASA LAS server](#)

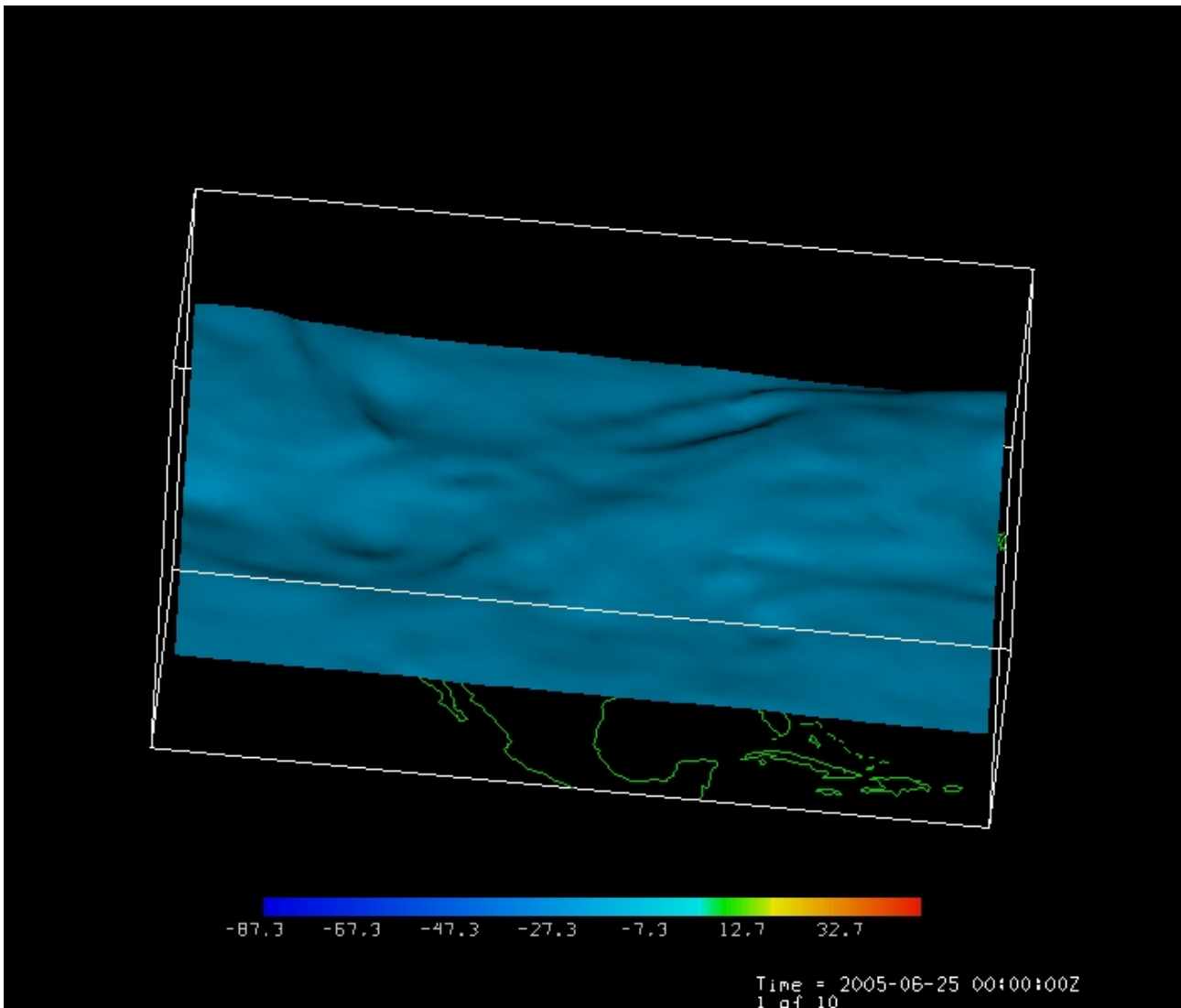


Illustration 1: Unidata IDV showing air temperature

Integration of OPeNDAP services with other data applications

As well as the data sharing capacities of OPeNDAP, Unidata has taken the trouble to develop a wide variety of programming libraries which allow access by common research tools. C++ and Java API libraries exist for OPeNDAP, designed to mimic or extend file-based NetCDF functionality, and plug-ins are available for packages like Matlab and IDL.

Prototype System Design And Assessment

When looking at the theoretical potential of a new technology, it can be useful to see how well that

theory stacks up in a real implementation. While formal performance testing has not been undertaken for what is still a research project, many of the hurdles overcome so far can speak to both the benefits and disadvantages of this system.

To assess the efficacy of OPeNDAP in the Bluelink project, several indicators are discussed. Firstly, a description of how the prototype system was built is given. Following this, the system is assessed according to the observable benefits, any outstanding issues, and what the current expectations are for the final system.

Building the Prototype

The primary difficulties have been integrating all data sources with the OPeNDAP server, performance issues and automating the publication of data sets.

The design of the system has also been heavily influence by certain specific requirements of the project. A cornerstone of the Bluelink project is the Meteorological Archive and Retrieval System (MARS), and in light of this some attention must be given to data storage issues.

...Data storage

The MARS system is a scientific database developed by the European Center for Medium-range Weather Forecasting (ECMWF)⁹ for the purposes of archiving large quantities data. It has an efficient retrieval time, and its query language supports some scientific intelligence, allowing a certain amount data manipulation to occur on the MARS server itself – much in the same way that OPeNDAP will allow data manipulation of its data on the server side.

⁹ <http://www.ecmwf.int/>

As a solution for storing large amounts of data while providing speed of retrieval, MARS is an excellent solution. MARS, however, does not have wide data format compatibility, nor good network functionality. As such, both storage and retrieval have posed a challenge for data formatting and access insofar as dealing with the native formats understood by MARS have been a challenge.

...Data formatting

Collecting data from many sources involves learning to cope with many formats. Storing this data either requires a database flexible enough to support many formats, or pre-processing to convert data prior to storage.

The data sources we have to deal with present some difficulty, although many data sources are easily handled. New data sources come in three major formats – NetCDF¹⁰, BUFR¹¹ or GRIB¹². Most sources of data tends to be available as NetCDF. Observation data is mixed, but notably in the cases of satellite and ocean profiles, arrives in BUFR format. For storage purposes, NetCDF is readily convertible to GRIB, and BUFR is natively recognised. For sharing purposes, BUFR data presents some difficulty.

...Sharing and Publishing Data

OPeNDAP development occurs in a world where data is exchanged chiefly in NetCDF. The resulting design has meant that while the architecture of OPeNDAP is flexible enough to support extension to arbitrary data types, the available implementations best suit only a relative few. The scientific community is forming into islands of standardisation, with particular groups adopting working standards, but not yet reaching universal agreement.

10 <http://my.unidata.ucar.edu/content/software/netcdf/index.html>

11 <http://www.wmo.ch/web/www/WDM/Guides/Guide-binary-1A.html>

12 <http://www.wmo.ch/web/www/WDM/Guides/Guide-binary-2.html>

One might question the lack of good sharing mechanisms for the MARS system. Interaction with MARS and the exchange of data is by means of compiled binaries executed from the command-line. Client tools exist, but rely on the MARS server being accessible within the same network. They function well, but are not secure enough for providing data externally. A web-based retrieval system does exist, but for various reasons it is not amenable to automation. In its home environment, there are many software tools, but these depend very much on the wider adoption of U.K. software solutions, which do not always meet our requirements. Additionally, they are a closed-source technology, which all else being equal increases the cost, and more importantly the dependence on continued maintainance of these products. For the purposes of interacting outside of our own network, software based on open standards is preferable.

In brief, the solution to providing data stored in MARS via OPeNDAP is an intermediate data-conversion layer. The hybrid system allows us to hide the implementational specifics of our back-end systems, allowing interoperability with external users of our data.

Unfortunately, data conversion is not possible for every source. This has resulted in a mixed system, largely split between model data and observational data. BUFR data sources have not proven straightforward to supply via OPeNDAP, so clients unfortunately need to work outside the new system in some cases. There is ongoing effort to address this.

...The OPeNDAP Server

The particular OPeNDAP implementation chosen was Thredds¹³, chosen for its particular flexibility when sourcing data. Alternative offerings are tailored for filesystem-based data sources, and as such are inappropriate for integration with potential intermediate systems of data delivery.

¹³ <http://www.unidata.ucar.edu/projects/THREDDDS/Overview/Home.htm>

MARS data sets are made available to Thredds via the intermediate layer in the form of a secondary web-server. This provides files via HTTP, populated with an index to known MARS requests. Initially developed in Java, performance issues forced a move to a similarly implemented Python cgi solution. The final solution was such that the OPeNDAP server could resource a dataset, presented as NetCDF and thus compatible with existing server implementations, from the intermediate Python broker, which would then perform the database request via the binary client on local disk.

Anyone implementing a similar system would be well advised to note two important aspects of this. Firstly, the Thredds server makes use of `getRange` http requests. While most web servers will honour these, the default API interfaces do not handle this for you. Secondly, the `doHead` method is also used. Once the `doGet` method is properly handled, despite the assurances given by some language documentation, one must also control the `doHead` behaviour.

Moving from Java to Python, while other changes to the algorithm were made in the process, saw an improvement of about a factor of 40, to 40Mb/s over the local loop. This was sufficient to remove the bottleneck.

...Securing the System

Securing the system is done through standard server mechanisms – which is to say that it is done through the servlet container's security model. In our case, this was the Apache Tomcat server, which supports many methods of authentication and authorisation.

This is not the place for an exhaustive discussion of the potential security models afforded by particular containers, but in principle container based security allows OPeNDAP to integrate with the existing security policies of the organisation. The prototype system works on this principle.

Observed Benefits

The observed benefits of the new system relate to the interoperability framework identified earlier. OPeNDAP has provided advantages beyond simple file exchange, simplifying the publishing of data, and allowing existing applications to easily use the new products. Specifically, the server has made inroads into the Format and Structure layers, with potential to eventually provide capabilities for the Semantic Structure and Metadata layers also.

The best example of this was the immediate ability of a number of clients (internal and external) to be able to access project data inside their applications without additional software development efforts. By adopting OPeNDAP, we at the same time met many of the requirements for application compatibility and integration.

The other benefits of the system have been a workable solution to providing networked access to our nonstandard database back-end, savings on network traffic, and a solution which should serve many projects going forward.

Observed Problems

There have not yet been any major problems identified with OPeNDAP itself. The difficulties encountered have all related to systems which are “behind the curtain”, which is to say they relate to providing data to the OPeNDAP server, rather than in interaction with the working system.

Expectations / Success Criteria

It is expected that OPeNDAP will provide a good interface for the exchange of the majority of Blueink products, in the form of gridded prediction fields.

Preliminary trials have indicated that the OPeNDAP server is capable of serving data at the underlying speed of the network, eliminating any concerns about bottlenecking.

Response times are in the order of several seconds, which is an improvement over existing systems, but this is an indication only.

Going Forward

OPeNDAP is a developing technology. Future plans outlined on the mailing list and on the web include increased use of XML as a metadata standard. It is interesting to see the scientific community pushing for the adoption of the same standards that are becoming entrenched in business computing.

The potential of greater compatibility is not to be under-rated. Many organisations are becoming increasingly market-oriented, with many of their budgetary mechanisms now moving towards project-based accounting. To make their products more saleable, it is desirable to adopt the standards already used by private industry. Being able to standardise on the delivery mechanism, especially if that mechanism can be the self-same one used by the organisation internally, provides obvious benefits.

...The OPeNDAP Community

Further information about the OPeNDAP project may be found at the OPeNDAP website. The

Thredds implementation adopted by the Bureau is developed at Unidata, a working group of the University of Carolina, and is under active development.

The development group at Unidata has been incredibly helpful and supportive. Requests for information and programming advice have always been replied to promptly and thoughtfully.

General inquiries should be directed to the OPeNDAP mailing list. There is, as yet, no formal group that has come together looking at the integration of this technology in Australia, but there is good reason for people to work together.

...Best Practises (Implementation Suggestions)

Getting the most out of an OPeNDAP solution is still somewhat of an art. Implementation is still largely done by software groups attached to particular projects rather than being fully integrated into the core IT systems of the organisation. Within the Bureau, its status as an important access path to new scientific data will bring it very much into the light.

Individual researchers wishing to make use of data from any particular institution can most simply do so by making use of any of the OPeNDAP-aware clients listed earlier, installing the OPeNDAP extensions available for many popular tools, or by making use of the C++ and Java libraries for accessing OPeNDAP files.

There is no centralised mechanism for finding out about specifically OPeNDAP data sets, but there is a list of catalogs available at the OPeNDAP.org¹⁴ site. Additionally, the National Virtual Ocean

Data System uses OPeNDAP technology¹⁵. The Bureau of Meteorology OPeNDAP server¹⁶

14 <http://www.OPeNDAP.org/data/datasets.cgi?xmlfilename=datasets.xml&exfunction=none>

15 <http://www.po.gso.uri.edu/tracking/vodhub/vodhubhome.html>

16 <http://dods.bom.gov.au/>

currently makes available only a few data sets. As the volume of data available via OPeNDAP increases, there may be opportunity for providing either a centralised community information server.. Until there is an appropriate way of automating this, community effort will have to go into publicising the availability of data via OPeNDAP.

Even largely “client” institutions can make gains by having an OPeNDAP server in place. Each OPeNDAP server can be thought of as much like a web proxy – it provides caching of both incoming and outgoing requests. In the case where the organisation may have many users for a particular data set, it can cut down significantly on the overhead by installing a local OPeNDAP server, configured with the data sets desired by researchers.

Closing Remarks

Implementing an OPeNDAP solution is one good way to manage issues of data exchange between institutions. It has particular advantages for certain scenarios, especially where the potential for reducing data volumes is high. Also, the benefits of having an extensible solution for publishing many data types in a unified way can be significant for interoperability.

The value to the research community lies in its ability to aggregate and disseminate scientific data from a wide variety of sources in a standard way. OPeNDAP has the potential to improve the nature of cross-institutional data exchange, and it is hoped that it will be enthusiastically adopted.