DHPC-037

Web-Based Access to Distributed High-Performance Geographic Information Systems for Decision Support

P.D. Coddington, K.A. Hawick and H.A. James

Distributed and High-Performance Computing Group
Department of Computer Science, University of Adelaide
Adelaide, SA 5005, Australia
Tel +61 8 8303-4519, Fax +61 8 8303-4366,
Email {paulc,khawick,heath}@cs.adelaide.edu.au

1 June 1998

Abstract

A number of applications that use GIS for decision support can potentially be enhanced by the use of high-performance computers, broadband networks and mass data stores. We describe our efforts in integrating together these powerful resources to provide useful services to Web-based clients for applications including environmental analysis for agricultural planning and land management; analysis of satellite and photo-reconnaissance images for national defence; and emergency services planning and response.

Many decision support applications that manipulate spatial data involve operations on very large data sets internally, which may require the use of high-performance compute servers, but carry out data reduction operations to provide summarised information to the end user. We have investigated the consequences of connecting high-performance resources together with broadband networks, but providing user access at the modest network capabilities of a modem link via the World Wide Web.

A range of technologies exist to manipulate spatial data and to integrate modules together. We focus on Web, Java and CORBA technologies and discuss the issues we have encountered in implementing prototype distributed geographic information systems incorporating "active" digital libraries, which provide remote data processing as well as remote data access. We have experimented with large online image repositories such as for earth observation satellites, as well as other sources of geospatial data.

Client/server computing, particularly over wide-area networks, is not yet widely used for GIS applications and research, but we believe it has great promise in systems such as we describe.

Keywords: distributed GIS, digital library, geospatial data archives, decision support

1 Introduction

In this paper we discuss the issues involved in implementing a distributed system that provides a computational infrastructure for the development of decision support and research applications requiring access to and manipulation of large geospatial data sets, such as satellite imagery, from remote servers. We give an overview of some prototype systems we have developed, and some more advanced systems that are still under development, which employ standard Internet and World Wide Web client/server technology.

The amount of digital geospatial data available is rapidly growing. In particular, there is a vast amount of data from earth observation satellites, and next-generation satellites are expected to produce terabytes of data per day. This presents a challenge for the development of computer systems that enable the storage, management and dissemination of these huge data sets in online data archives or digital libraries. Ideally, such a system would provide efficient, on-demand remote access to these data sets over the Internet (or an intranet), so that authorised users could easily access and utilise the data for a variety of Geographic Information Systems (GIS) applications, including decision support, research and other analysis.

For a number of GIS applications, such as those requiring real-time or interactive analysis of large data products such as satellite imagery, the processing requirements are large enough that high-performance compute servers are required. This leads to the concept of an "active" digital library, where the server provides not only services for querying and downloading of data from the library, but also services for processing the data before downloading [1, 2, 3, 4, 5, 6].

This approach is particularly useful if the amount of data to be processed is very large, for example multiple channels of a satellite image, but the final result is relatively small, for example a processed satellite image for a localised area, or perhaps just a few numbers such as average sea temperature or percentage cloud cover or some correlation coefficients. If the data is obtained from the server using a wide-area, relatively low-bandwidth network, it will be more efficient if the user only has to download the final results rather than download the large input data set and process it locally. Many decision support applications that manipulate spatial data involve operations on very large data sets, but carry out data reduction operations to provide summarised information to the end user. Some of the data sets may be remotely accessed from different servers, possibly over wide-area networks, and the processing may be done on yet another machine, possibly a high-performance computer or supercomputer. We have investigated the consequences of connecting together resources for fast mass storage and high-performance computing with broadband networks, whereas the user's client computer may only have modest network capabilities, such as a modem link via the World Wide Web.

We describe our efforts in integrating together high-performance computers, broadband networks and mass data stores to provide useful services to Web-based clients for decision support applications including environmental analysis for agricultural planning and land management; analysis of satellite and photo-reconnaissance images for defence command, control, communications and intelligence (C3I) systems; and emergency services planning and response. Many of the end-users of this technology are in a similar situation – they are gradually making the transition from traditional decision support systems based on manual processing of non-electronic data such as maps, aerial photographs and TV weather reports, to information age decision support systems based on automated or semi-automated processing of digital data from large online data archives.

We have experimented with large image repositories such as for geostationary meteorological satellites, as well as other sources of geospatial data such as digital terrain maps. Operations on bulk data range from simple data overlays, to computationally intensive operations such as data interpolation and registration and rectification. Some of these operations are infeasible to do in real-time except on supercomputers. The client programs that can set up processing on demand of various data sets need to be robust, and yet easy to use with suitable graphical interfaces.

In section 2 we discuss mechanisms for enabling an active digital library, using as an example our online repository of geostationary satellite data and some prototype systems for providing access to this data and services for processing the data. We discuss the issues we have encountered in implementing prototype systems using Web, Java and CORBA technologies. Section 3 describes some of the software and hardware components that are needed to implement a large-scale distributed geographical information system based on the active digital libraries concept, and the middleware that is needed to integrate these components into a robust and efficient infrastructure that can be used by a variety of applications. In section 4 we present some general scenarios illustrating how decision support systems can benefit from the use of fast on-demand access to archives of geospatial data and distributed GIS services, with more specific applications being described in section 5. We briefly summarise our findings and conclusions in section 6.

2 Online Geospatial Data and Services

With the advent of cheaper and more powerful computers, networks and electronic storage media, and particularly the huge increase in the use of the Internet and the World Wide Web, large online data archives and digital libraries that can be accessed over the Web are becoming widespread in both the commercial and scientific arenas. The technologies for implementing and accessing such archives, are for enabling online data processing, are still evolving. Our On-Line Data Archives (OLDA) Program [7] is investigating these issues, particularly as they relate to large satellite data archives.

There are a number of existing projects aiming to provide access to digital libraries of earth observation data. These include the Synthetic Aperture Radar Atlas (SARA) at Caltech [6, 8]; NASA's Earth Observing System Data and Information System (EOSDIS) [9]; the European Space Agency's online product catalog [10]; and the the Australian Centre for Remote Sensing (ACRES) Digital Catalogue [11].

Like these and many other projects, we are investigating the computer systems and software issues related to the efficient storage and dissemination of data in large distributed online archives, or digital libraries, of satellite data. However our work, and similar efforts such as the SARA project, is also focussing on developing active digital libraries, which enable remote data processing as well as remote data access. New technologies such as the World Wide Web, Java [13] and the Common Object Request Broker Architecture (CORBA) [14, 15] are making it easier to develop portable distributed client/server systems of this kind.

We are implementing our software support infrastructure using a set of server-side programs accessed using a mixture of Java programs communicating between client and server as well as between servers; CORBA for invoking remote procedures through an object-oriented interface; and the Common Gateway Interface (CGI) [12], the standard mechanism for invoking processes on a Web server. The client is a customised Web-based interface using Java applets [13] that can be downloaded and run from a standard Web browser. This architectural mix provides a portable and powerful framework for rapidly prototyping systems that can integrate existing server-side utilities, and is extensible to create complex systems with the desired functionality. Most importantly, it is built using Web standards, with standard Web browsers for the clients.

2.1 A Prototype Active Digital Geospatial Library

Our prototype system implementing an active digital library for geospatial data is known as ERIC [4, 5]. This is a Web-based interface to a repository of data from the Japanese GMS-5 geostationary meteorological satellite. This satellite generates 4 images (one visible band and 3 infrared bands) plus associated metadata every hour, which is approximately 200 Mbytes per day, or 75 Gbytes per

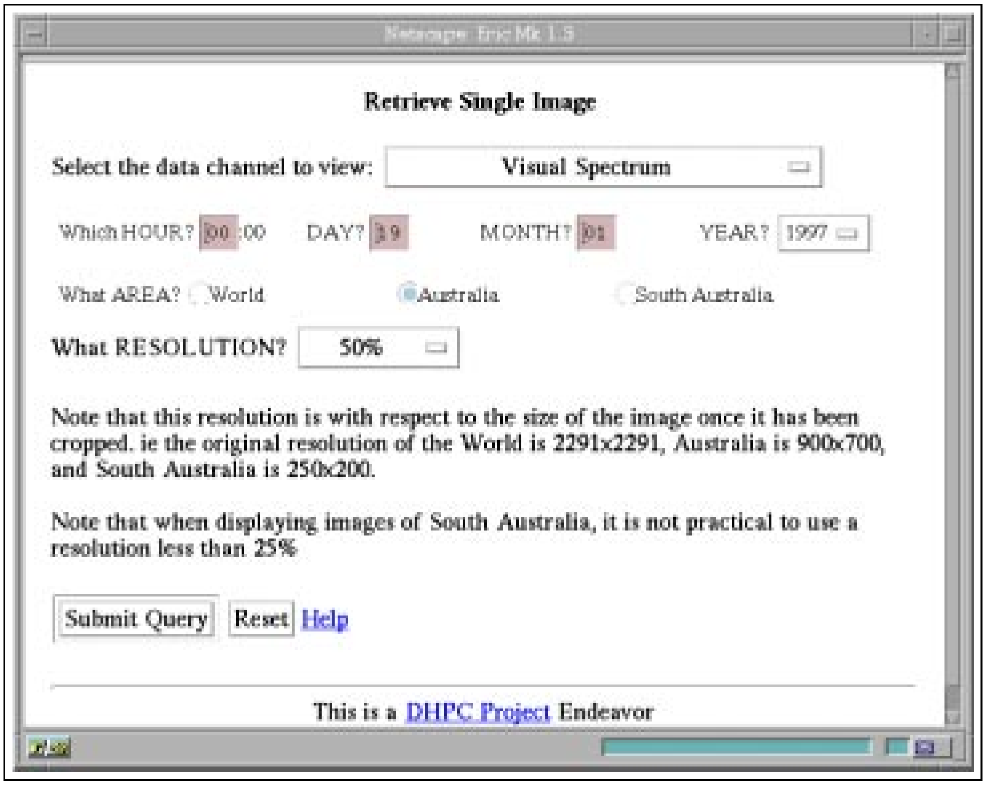

Figure 1: ERIC: Single image query form.

year. The data is stored on a combination of a RAID disk array and a tape silo, which are accessible to the server by a high-speed (155 Mbits/sec) network.

Our initial implementation of ERIC allows searching the GMS-5 data set by querying metadata fields such as date, time, and spectral band. The user can view thumbnail images for products matching the query, and then download the image data and associated metadata. In addition to these standard digital library functions, ERIC allows the user to invoke some services for processing the data on the server and to download the processed data. The user can specify a spatial region, such as Australia, which is cropped out of the image, so that only the image data for the requested region is downloaded. The user can also request that the data be provided at a lower resolution, and/or using lossy compression (such as JPEG), for faster downloading. A time series of images for a particular region and resolution can be specified, and the server will process them to create an animation, returning the result as an MPEG (digital video) file. Some of the ERIC interfaces, for selecting and displaying a single band satellite image, are shown in figures 1 and 2.

As an example of providing a more complex data processing service, ERIC can also compute cloud cover for an image. This is done by an algorithm that identifies cloud as cold, bright pixels, which requires using both the visible band (for brightness) and an infrared band (for temperature). Other processing could be incorporated into this system, for example geo-rectification of the data.

The initial implementation fo ERIC has a fairly simple user interface using HTML forms within a Web browser, which invokes programs on the server using the Web standard Common Gateway Interface (CGI) [12]. The server consists of CGI shell scripts and Perl programs that manage a number of different programs for data processing. For example, in computing the percentage cloud cover over a particular region, the script needs to invoke processes that interpret the request; find

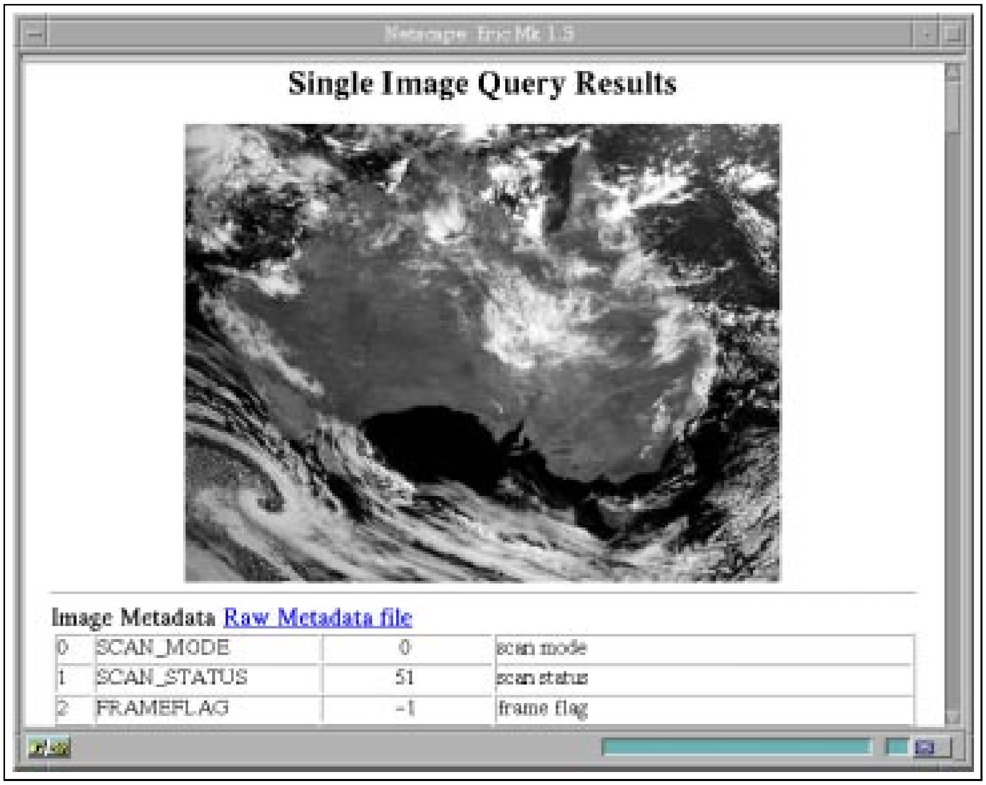

Figure 2: ERIC: Single image results screen.

the appropriate data files that match the request; copy the files to a temporary working directory; uncompress the visual and infrared data files; extract the image data pixels from the particular image file format; crop out the requested region; process the image data to identify areas of cloud; compute the percentage cloud cover; return the result to the client; and remove or cache any temporary files.

An important functionality provided by ERIC is the caching of results. For example, many queries may request geo-rectified data for the whole of Australia. Since it is computationally expensive to generate this data, and it is likely to be reused, it is worthwhile to cache the data if there is adequate storage available. Managing the cached files is a complex problem, and requires routines for selecting what products should be cached, for naming and rediscovery of the cached files, and removing the least-requested files when storage becomes limited.

Using an interface based on HTML forms and CGI is adequate for simple applications, but it severely limits the functionality of the client. A better approach is to use a Java applet for the client, which allows the development of a sophisticated user interface, as well as the ability to do some processing on the client where it is advantageous (for example, manipulating a graphical user interface), rather than all the processing being done on the server, with the accompanying problems of potentially large latency and low bandwidth communications.

2.2 Standards and APIs

In order to make Web-based distributed geographic information systems more general, so that the clients and servers can handle any kind of geospatial data set, and to develop distributed systems with multiple clients and servers that are interoperable, there are two important standardisation issues that must be addressed.

The first is the standardisation of metadata, so that searching digital libraries by querying metadata can be done using the same metadata fields for different data sets. There are several organisations working on metadata standards, both for specialised fields such as geospatial data [23, 24, 25], and in general [21].

The second issue is the standardisation of the client/server interface, so that different clients and servers can be interoperable, and clients can request data from multiple repositories using a standard interface, or an application programming interface (API). The API to an active digital library should specify mechanisms for querying, processing, and retrieving of data. Using general metadata standards, it would be possible to construct a basic general API for querying of metadata and downloading of data that would be universal across any kind of digital library and any kind of data (general Web search engines do this for restricted data sets, such as Web pages and newsgroup postings). However to allow more powerful and domain-specific queries, and to provide an interface to data processing services which would in most cases be specific to the application and type of data, specialised APIs are needed that are specific to a particular domain or topic — for example, the services and interfaces for an active digital library of geospatial data would be different to those for a library of human genome data.

To develop a more advanced active digital library for geospatial data, integrated with an improved version of ERIC, a well-defined API is required. As with metadata, proposed standards are gradually being developed for digital libraries and data archives. There are various groups, including the Open GIS consortium [22], the International Standards Organization (ISO) geographic information technical committee [23], and various government agencies in the U.S. and elsewhere [24, 19], that are all involved in developing standards for the storage and exchange of geospatial data.

Currently the API that appears to be the most advanced in design, and the most suited to our existing archive of GMS-5 satellite data, is the Geospatial and Imagery Access Services(GIAS) [16] specification, which is a fundamental part of the U.S. Imagery and Geospatial Information System (USIGS) [17, 18] developed by the U.S. National Imagery and Mapping Agency (NIMA) [19]. This system is targeted at use by the U.S. military, for analysis of satellite and photo-reconnaisance imagery, however it is general enough to support the storage, management and dissemination of geospatial imagery data for a more general distributed GIS. We are therefore developing an active digital library for GMS-5 satellite data that is based on the GIAS. This prototype system could be generalised to support any kind of geospatial data.

2.3 Implementation of a Standardised Geospatial Data Archive

The GIAS specification describes an object-oriented archive management system for an active digital library. The specification is structured so that the main functions of the library, such as adding data, querying the metadata, and downloading the data of interest, are controlled by different *managers*, each of which is handled by a different class. Remote access to the system is implemented using the CORBA standard, and described using Interface Description Language (IDL) [15]. The GIAS also defines a standard Boolean Query Syntax (BQS) which specifies a format for metadata queries on geospatial data. More detailed information on the GIAS can be found in the specification [16].

We have developed a prototype implementation of a GIAS-compliant geospatial digital library [20]. This work was done in collaboration with the Imagery Management and Dissemination (IMAD) Project group of the Australian Defence Science and Technology Organisation (DSTO), who are implementing a prototype distributed system for handling photo-reconnaissance imagery for military command, control, communications and intelligence (C3I) applications. The same system can also be used for a variety of non-defence GIS applications, such as those described in section 5.

We have initially implemented a subset of the GIAS that provides the basic functionality for managing a geospatial image library. The implementation includes the main managers for the server, an interface to a database on the server for storing the queryable metadata, a translator that converts BQS queries to standard SQL database queries, the remote invocation of the managers using CORBA, and a client for testing the complete functionality of the subset of the GIAS that we have implemented. Figure 3 shows a screen image of the test client. Only the queryable metadata is stored in the database, while the imagery data is accessed via the standard file system, and may be stored either on disk or in a tape silo.

The GIAS provides a sound basis for implementing an active digital library of geospatial data. It specifies interfaces to the basic functionality of searching for data products by querying metadata, and downloading the required data in a specified format and resolution. It also provides interfaces for some basic data processing services of the kind provided in the original ERIC system, such as cropping out a specified geospatial region, and creating animations of time series of images. The design, which is based on CORBA, is capable of supporting additional data processing services, however an API would need to be defined for each one, and included into an extended interface specification.

We are currently working on interfacing our existing GMS-5 repository to the GIAS server. This requires loading metadata for all the existing GMS-5 data into the database used by the GIAS implementation, and converting our automated data ingest programs so that new data from the satellite is automatically incorporated into the GIAS system.

The GIAS implementation provides most of the services required to support the functionality of ERIC. We are developing a new, GIAS-compliant version of ERIC, using a Java applet rather than HTML forms as the user interface, connecting to the server using CORBA rather than CGI, and implementing the server programs using Java processes which can call other programs as native methods, rather than calling them from shell scripts and Perl programs. Using CORBA requires an Object Request Broker (ORB) running on the client as well as the server, however CORBA and the Internet Inter-ORB Protocol (IIOP) [14, 15] that it uses are now being integrated into Web servers and browsers, for example the latest versions of Netscape Communicator feature a built-in Java ORB to handle CORBA requests [26].

By implementing the ERIC user interface as a Java applet, it can easily be customised for different applications, which can download the basic applet, plus classes specific to the particular application. Using Java programs on the server allows for a more complex and modular system than could be constructed from shell scripts and Perl programs, and allows much better error handling, which was a major problem with the original version of ERIC. The GIAS defines standard exceptions (errors) as part of the specification.

3 Middleware Issues

The GIAS, and even the USIGS, does not specify a complete architecture for developing an imagery management and dissemination system, although they provide the basis for such a system. The GIAS provides a specification for remotely interfacing to a single image library, however in a real system, there may be many distributed libraries. The client has to know how to find a particular library and be able to search over multiple libraries. This is the function of a trader [14, 15], and is not specified by the GIAS or USIGS.

There are many other issues involved in creating a distributed digital imagery archive, particularly related to security and performance, that are also outside the scope of the USIGS specification. For example, if a library (or parts of a library) are mirrored at different physical locations, the client should be able to connect to the location which can provide the fastest response time. Also, most of the images in large image libraries will be held in a tape silo, which may have access times on the order of a few minutes. Mechanisms are needed for implementing strategies such as accessing

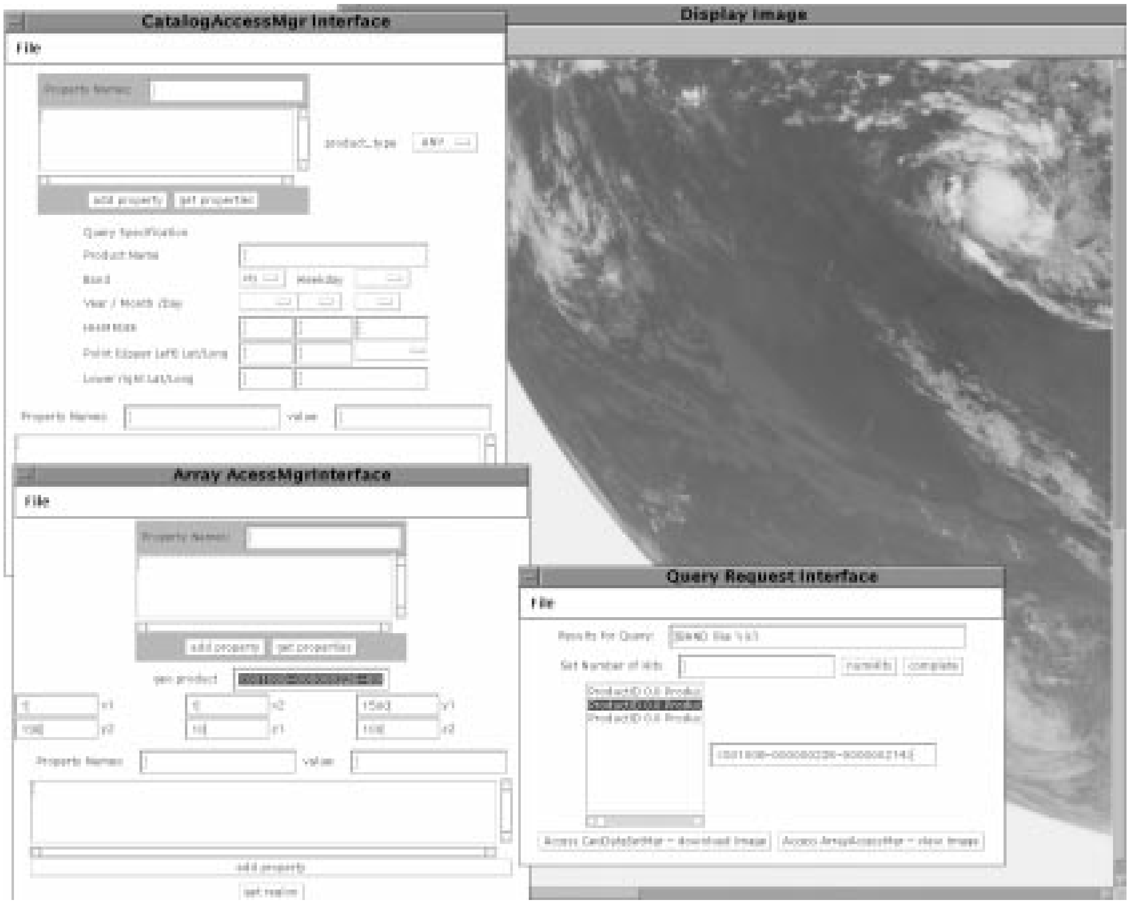

Figure 3: A screen dump of some manager windows from the test client for the GIAS implementation. In this case, a query has been made using the Catalog Access Manager, a section of one of the products matching the query has been selected and downloaded using the Array Access Manager, and the image has been displayed in a separate window. The purpose of this client is just to test the functionality of the GIAS implementation – an application client would have a more user-friendly interface.

lower resolution images stored on disk, prefetching and/or caching images from tape, tiling and compression of images, etc.

These problems can be dealt with by a middleware system, whose job it is to provide an infrastructure for complex, distributed applications. Developing such middleware is a challenging problem. We are developing a middleware system known as the Distributed Information Systems Control World (DISCWorld) [1], for building distributed applications involving accessing and processing data in large online data archives.

A number of key issues must be addressed for a middleware environment to successfully manage the resources for distributed geographic information systems and decision support services. Firstly, the system must identify service and data components and deal with these in as high a granularity as possible, in order to reduce the problems of network latency and limited bandwidth. However, there is a tradeoff between low-level processes that will be very reuseable but hard to invoke except via a programming language, and high-level processes that are more efficient to run over a network and can more easily be incorporated into a global name space and invoked as Web-based services, but will be less reuseable.

Interoperability across multiple systems that were previously independent, and handling different data formats and system interfaces is an extremely challenging problem. Once the distributed system crosses organisational boundaries, there are additional issues of security and authentication that must be addressed.

The system may be built with data delivery networks that vary in bandwith. High-speed broadband networks can make a big difference in performance of a distributed system, and should be used where they can make a difference. In general, this will be to connect distributed compute servers and data archives, providing high-performance access within organisations and between vendors in the hierarchy who need to exchange bulk data. Customer end-user connectivity may be just modem level Internet access if relatively succinct decision support data is to be delivered.

The distributed system should also allow incorporation of high-performance and efficient hardware components, and use scheduling algorithms to make the most cost-effective use of available processing, networking and storage capabilities.

4 Distributed Systems for Decision Support

In this section we illustrate some typical scenarios for decision support systems that access distributed data archives across wide-area networks, and discuss the middleware and distributed computing issues which we believe are critical for their widespread uptake.

One example from the area of land and environmental resource management is particularly relevant to Australia, which derives a substantial part of of national productivity from primary industries. A great deal of spatial and land resource information is now available from various satellites, aerial reconnaissance, measurement study programs, and other sources such as census and land registry. There are many interoperability and ownership and legal access issues involved in multi-source data. Although these are a great impediment to wide-area distributed decision support systems, we do not address these issues here. Another significant impediment is the difficulties in setting up the software interoperability and machine access for decision makers to access and manipulate such data. We will concentrate on these issues, and in particular discuss the consequences of wide-area distributed computing and metacomputing in addressing these difficulties.

Imagine a station manager or farmer planning the year's activities and wishing to make sensible decisions regarding irrigation and rainfall runoff; crop rotation and planting; expected yield and optimal harvest times and other land care operations such as salinity management and prevention. It is unlikely that the farm manager is working without some existing base of knowledge. There is very likely a base of experience that may have been passed down in the family or is available from neighbouring stations or other individuals in a similar situation. Nevertheless, there may be newly acquired land or new situations, new crop species or irrigation techniques that expand the list of option available to the farmer. How can they exploit the data that may be available to aid the decision making process? Some data will be available in the form of highly sophisticated processed data products such as short and long term weather forecasts. These may be available at the resolution and localisation required for precision agriculture decisions, or may only be available in undigested form. Perhaps no-one has asked for detailed forecasts or data for the farmer's particular region before at the requested resolution. It may however be entirely possible for an automated processing system to be able to create the desired products. Aerial reconnaissance data can be readily bought for a particular region by hiring a suitable plane and aviator. Fly-over data at very high resolution from various commercial satellites is quite likely available in raw form for the region the farmer requires. How can a processing and automatic product creation framework be put in place to allow those organisations who do have the necessary skills to create the desired products to do so economically for what might be a set of isolated one-off sales?

A wide-area metacomputing environment built using a set of clustered computing resources can be set up to provide a common shared resource for the customers (farmers and land managers) to interact with the value-adders or organisations who can create processed products from raw data and the raw data suppliers or government custodians. What is needed is a suitable set of middleware or software that can provide the necessary interoperability and scheduling of the necessary computing services.

There is a feeding chain relationship that exists between the end user of a processed product, the custodians of the raw geospatial data, value-add processing agencies which may involve several intermediate stages, and the final data suppliers. In many cases around the world raw data belongs to the government and may be made available for the public good to anyone who wishes to use it. Government agencies typically have a dedicated and customised infrastructure for processing raw data to construct decision support products. Government operated weather bureaus are an example of this. The feeding chain model for this is shown in figure 4, where the entire system can have dedicated hardware, software and a high bandwidth data access system.

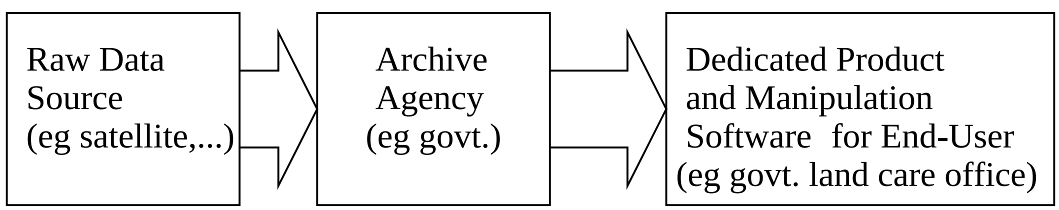

Figure 4: Dedicated decision support system for a large organisation such as a government agency (a government land management office or weather bureau for example) with its own customised, dedicated processing system for generating decision support products.

Many organisations now exist to sell products derived from this raw data, possibly involving considerable expertise and innovation as well as sophisticated processing equipment. The feeding chain model that allows vendor companies to value-add to existing data is shown in figure 4, illustrating how the end-user receives the final product through the vendor alone and need not be aware of the raw data such as satellite imagery that went into its construction.

Some value added products such as weather forecasts are by no means trivial to derive from raw observational data, and weather agencies around the world expend a lot of resources in assimilating data and producing their products. The economics of forecasting is selling to a large volume market,

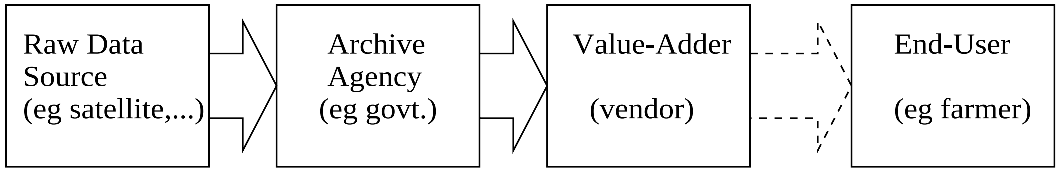

Figure 5: The feeding chain for vendors to value-add to publicly archived or government data to generate lower volume, higher margin data products for decision support.

since so many people and organisations want accurate weather forecasts that they are generally sold at a very low margin, if any at all. Land management decision support products tend to be more esoteric and require large margins to be worthwhile for a value-adding organisation. In consequence the decision support capability of such data tends to be underutilised. Many government agencies set up their own special systems (as shown in figure 4) to provide spatial or geographic decision support for their areas of jurisdiction. Weather agencies are moving towards more automation which will lead to a driving down of the cost of production of value-added forecasts, but automation is even more necessary for low-volume market areas.

The low-volume market can be effectively served by a hierarchy of different value-adders all interoperating, on-selling products derived from each other's work, and ultimately derived from the bulk raw data that may itself originate from satellite or other sources run by a commercially-operated company. This hierarchical relationship is shown in figures 4 and 4.

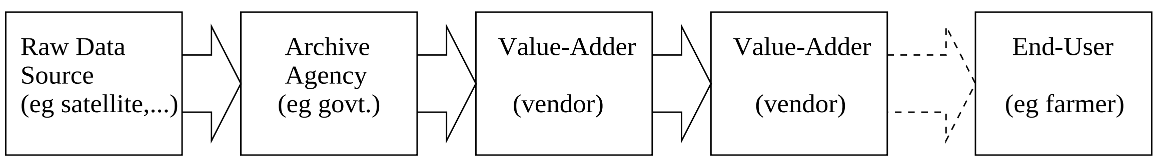

Figure 6: Hierarchical relationships between value-adders allow complex end-user products to be derived from a feeding chain of on-sold data products. Each value-adder on sells derived data products.

A number of general purpose computing technologies can be used to improve the automation of these value-added services. Improved data archive management facilities can provide faster bulk access to data sources more cheaply. A number of systems for archiving data make use of what is essentially a smart middleware to control a mix of storage media – many cheap bulk devices such as tapes are housed in multi-tape silos under robotic control, working in tandem with fast but more expensive devices such as bulk disk or RAID arrays of disks which can act as a buffer for frequently accessed data. The primary data suppliers may choose to use technology such as this in managing the bulk archives of raw unprocessed data. Value-adder organisations may require access to more than one primary source of bulk data simultaneously, as shown in figure 4, and it may prove economic for suppliers to use on-line archive technology such as combined RAID and tape silo systems to minimise the amount of data that needs to be replicated at various sites.

Processing resources can be pooled together by organisations to provide better response time or resource utilisation than if individual departments under-utilised their own resources most of the time just to be able to handle their occasional peak capacity requirements. In particular many organisations have clusters of workstations or personal computers that are only sporadically used by individuals during working hours and which stand idle overnight and on weekends and holidays.

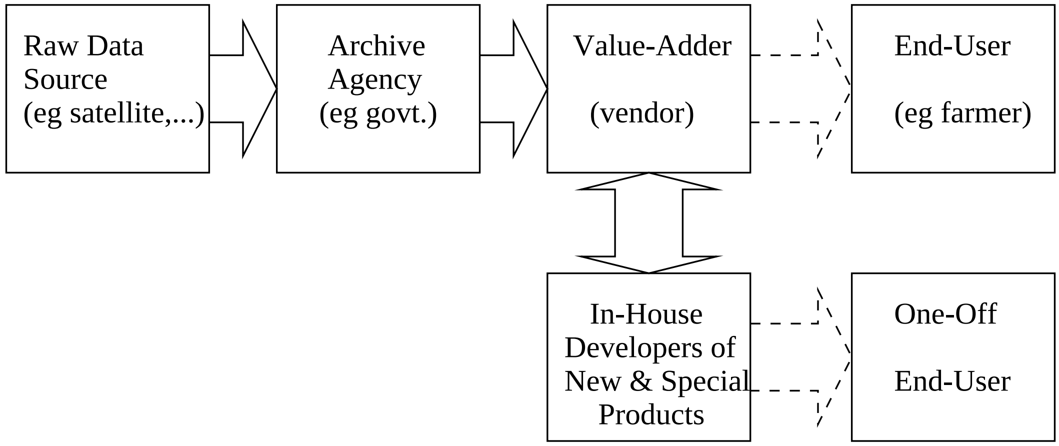

Figure 7: Value-adder combining primary data sources by data fusion into a complex output data product.

It is often too difficult to make use of these lost cycles due to lack of smart interoperable scheduling software environments. Improved scheduling on clusters can turn an unused resource into a virtual supercomputer for these organisations.

Storage technology can be integrated with World Wide Web data delivery techniques to provide casual customers access to a plethora of pre-prepared data products with very low shipping and delivery costs in spite of up to the minute relevance. Nevertheless public demand for network bandwidth is growing and better data connectivity between data suppliers and value adders is likely to be necessary to avoid competition with public Internet traffic. Smart middleware can be used to manage the existing networks between cooperating suppliers in a feeding chain, to make sensible use of data caches and intermediate storage so as minimise peak loads and hence conflicts on the networks. This idea is illustrated in figure 4 where a value-adder has some sort of smart store that can cache products according to some access policy and thus archive processed data. A smart middleware management system is capable of managing access pattern predictions and prefetching data for certain applications and users. This would make use of a network of caches and archives to make optimal use of the customer data delivery network.

End-users of a particular value-adder can generate a homogeneous or heterogeneous mix of product requests. A particular vendor may benefit from the caching model shown in figure 4 even if the requests are heterogeneous in nature, providing the intermediate products, from which different final products are derived, are cached to allow fast reuse.

In addition to the normal feeding chain of products supplied by a value-adder to their customers, some degree of product development is likely to be ongoing as an in-house effort or perhaps in collaboration with special customers. These new or one-off data products may be researched and developed by special users who have access to the individual processing services inside the value-adder's middleware system. The decision support products thus developed may eventually make the transition to being fully automatable products for normal on-demand delivery to all customers. This model is shown in figure 4. The value-adding organisation's staff then spend their time developing new products and services, and encoding these into the smart scheduling environment as well as handling those occasional special jobs which require some manual intervention. The schedule management environment might still provide substantial help to the value-adders staff undertaking special jobs, as it can provide the primitive operations and support environment. We now have a mixed scenario with different levels of users all drawing different services from the environment.

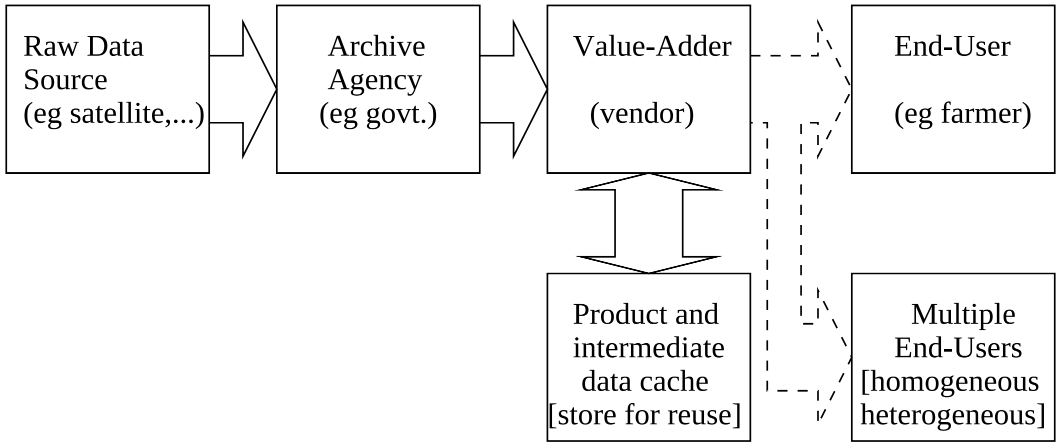

Figure 8: Homogeneous and heterogeneous mixes of end-user requests benefit from smart caching of final products as well as intermediate data used to generate final products.

In summary, better middleware to manage storage of raw data, exploit under-utilised processing capabilities such as existing clusters of workstations, and improve data network delivery using mechanisms such as smart caches and prefetching, can significantly enhance the performance and cost performance perceived by decision support users. This is particularly effective over large wide-area clusters of resources, where one might expect the statistical load fluctuations from individual users is balanced out.

A number of middleware products have already been attempted and several techniques are being presently researched. An area we are addressing ourselves is that opened up by restricting the general scheduling problem to that of handling only well-defined high-granularity services.

Consider the types of services that might be required for the land care and management decision support scenario we have outlined. These are not arbitrary programs running with arbitrary data. Instead they are generally drawn from a subset of well-defined application components with known performance requirements and will typically be run on a very restricted set of data sets and data set sizes. Consequently it is possible to set up a scheduling management system that has access to the characteristics of each application component and can therefore predict and hence optimise more accurately the cost and time needed to carry it out on a set of computing resources. The smart scheduler is therefore able to make very good use of the whole set of resources under its control and can organise user requests in priority, running the most important or urgent on fast resources, and the lower priority actions using slower, cheaper queues. Given that a restricted finite or manageable set of applications and data sets are being used under the system as a whole it is also possible to effectively cache or store frequently accessed or requested data products.

Consider an example. Suppose a farmer has requested information regarding the prediction of crop yield and optimal harvest time. Crop acreage may already be known to the farmer, or can be accessed from either a land registry database or perhaps calculated from a satellite image which is geo-rectified and registered to allow a geographical area calculation. Localised weather data for the current growing season, weather predictions for the region, and weather data and crop yields from previous years may be combined to calculate an estimate of the likely crop yield. A practical system might provide a series of possible harvest time predictions and options with the computed consequences for the farmer. The decision support product delivered to the farmer may only be the summary output of the calculations – a few kilobytes perhaps, whereas the raw data upon which the calculation was based may have represented many gigabytes of spatial imagery, digital

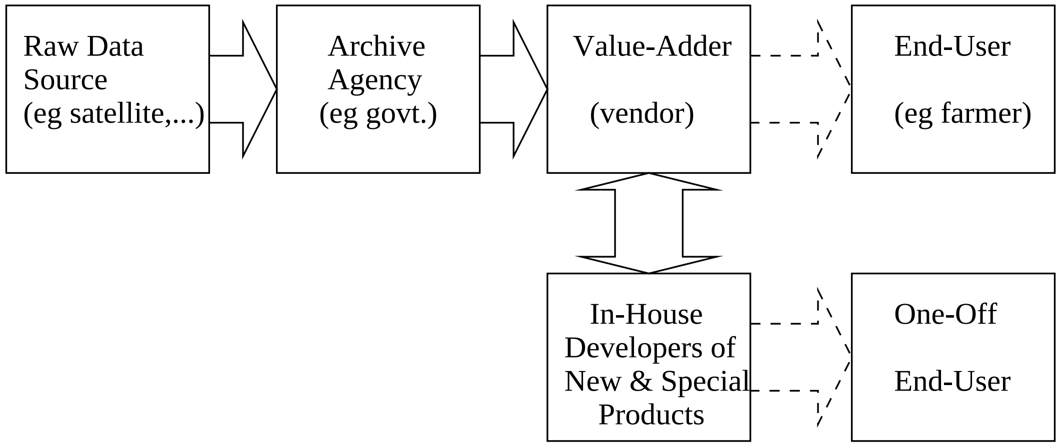

Figure 9: New and one-off special products can be developed by in-house special users, for eventual automation as on-demand end-user products.

terrain data, runoff patterns and so forth. The final product can therefore be easily delivered over a low-bandwidth network to a client application running in a Web browser.

Similarly, the computational power of the client computer the farmer might use to place the query and request the decision support product might be low, whereas many processing cycles may have been used by the value-adding organisation in creating the data product. Suppose the farmer makes a similar request the next day, but perhaps with some additional new information. By controlling and caching the intermediate data products the value-adding organisation used to create the product such as the raw satellite imagery for the farmer's particular region, it may well be possible to save considerable re-processing time. The profit margin on the second product will be much greater - or of course the value adder may choose to pass on the margin saved to its customers in the form of a cheaper price. This model of being able to cache intermediate data products may provide significant savings and better utilisation of resources. It can only be set up under a smart caching system however. It would be too hard to track the necessary data items manually.

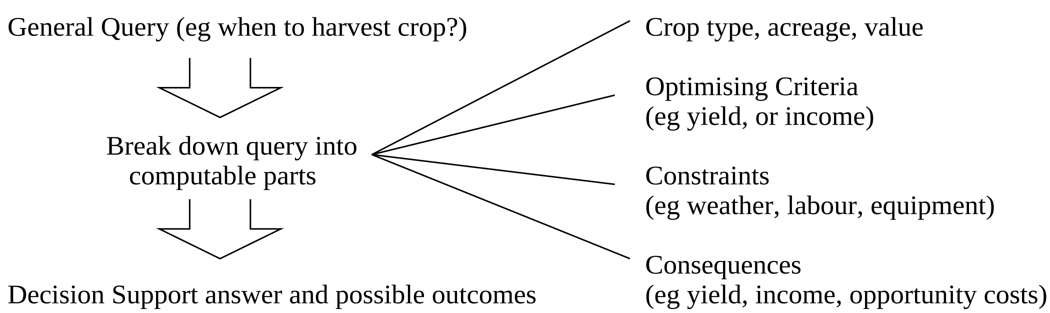

Figure 10: Decision support possibilities if a complex query can be broken down into a set of computable parts.

The farmer's query is decomposed into a sequence of well-defined and computable steps in the example above. This is shown in figure 4. Each of these steps might be an application component with a well-characterised performance that can be controlled by the smart scheduler. The value-adding organisation that provides some particular services and data products might have access to archived raw data either as a locally archived copy or from some other data provider such as a government agency or another vendor. Our value-adder might be able to respond semi-automatically to the incoming query and deliver the product and a bill to the farmer using a suitable scheduling environment which manages its computing resources.

It is not always entirely possible to automate product generation and delivery. Some products will always require some human expertise or intervention to create them. A common example is the detailed local weather forecasts issued by government weather agencies. A great deal of automatically collected raw data such as satellite imagery, and computer model generated data is used as input for regional forecasts, but the complexity of the problem is such that these data products are generally only used as input to an expert forecaster who digests the data and uses them as decision support information in producing his own actual forecast. The forecaster has final say on the final product, but has made use of a highly sophisticated decision support environment in producing it. This human intervention stage can occur anywhere along the feeding chain, and the middleware support environment can be thought of as a framework to support those humans involved.

5 Decision Support Applications requiring Distributed GIS

Other decision support applications share many of the properties of the scenario we have outlined above. Here we present a brief description of some of the specific applications that we are targeting with our prototype distributed GIS software infrastructure and active digital libraries.

5.1 Land Management and Crop Yield Forecasting

An example scenario of providing decision support information for land management using a Web-based distributed GIS was given in the previous section. There are many similar applications. For example, as well as providing localised crop yield forecasting for an individual farm, there is a demand for crop yield forecasting on a regional and national level. This requires the use of large amounts of disparate data that may be distributed over many sources. If this data is made available online, it can be integrated by a distributed GIS, which can also provide support for including compute servers and interfacing to the software to perform the crop yield prediction.

A distributed GIS has obvious applications to large-scale land care studies and environmental monitoring, which also require large amounts of disparate geospatial data, including satellite data. For example, a researcher may want to combine vegetation index data obtained from NOAA satellite images with rainfall data obtained from weather bureau data and GMS-5 meteorological satellite images, in order to correlate rainfall with vegetation growth, to provide better understanding of the environment and better models for predicting crop yields, or providing localised predictions of rainfall and frost.

5.2 Defence and C3I

Defence applications such as processing of photo-reconnaissance images and other intelligence products can be viewed using a similar model framework to the scenario presented in section 3. Many intelligence products originate and from satellite or aerial reconnaissance flights and are archived using various technologies including digital data systems. Various organisations within a government's defence forces may collect and archive their own data and may make it available to each other. Sources may vary in quality and type considerably. Some organisations may work as in figure 4 in an entirely in-house customised system, but more frequently economic and interoperating reasons require various value-adding relationships to be set up with the whole defence force to share data and to derive intelligence products from multiple distributed sources and 'on-sell' derived products for decision support. Human analysts may be working as end-users or as value-adders in the system itself, combining data products into decision support material for those processes that are not yet automatable. The defence community has some additional constraints but overall the model is very similar to that for commercial value-adding of land resource data.

Some additional properties however include: real-time or near-real time data delivery; tighter security and secrecy between components of the system; encryption and restrictions to certain parts of the 'market'; and closed rather than open product catalogues. In spite of these differences the quality and quantity of the data components and the automated and human enhanced processing activities still conform to the decision support information delivery model outlined in section 4.

We are currently working with the Australian Defence Science and Technology Organisation (DSTO) to develop a prototype Web-based imagery access system as a technology demonstrator [20]. The system is aimed at a variety of C3I clients, from image analysts using fast networks and high-end graphics workstations, to commanders in the field using laptops and low-bandwidth connectivity.

5.3 Emergency Services

There are two basic scenarios for the use of GIS by emergency services. The first is for emergency planning purposes, to investigate measures for alleviating or responding to emergencies such as fire or flood. These systems allow the user to simulate possible situations and their outcomes. For example, firefighters or forestry managers might run simulations of what would happen if a fire was started in specified areas under certain conditions, in order to identify danger areas where fuel loads should be reduced with cool burns, and under what conditions these managed burns would be safe. Emergency services might run simulations of evacuation procedures in the event of a flood or fire.

The second scenario is decision support during an actual emergency, which provides a much greater challenge to a distributed GIS. As with emergency planning, the GIS will draw on a wide variety of existing geospatial data, such as the positions of populated areas and houses, road networks, fire stations and hydrants, etc. However in this case, it will also require access to real-time data that may be rapidly changing. For example, a GIS for decision support during a bushfire would ideally have real-time data feeds coming from many different sources, including the positions of firefighters, police and emergency services personnel in the field, information about the position of the firefronts, which can be transmitted from helicopters with GPS units, and current and predicted weather information such as temperature and wind speed.

A Web-based distributed GIS could be accessed in the emergency response headquarters, as well as by crews in the field using a laptop connected by a cellular modem. The GIS could provide up-to-the-minute information on the situation status, and provide decision support information such as the optimal routing of fire trucks or evacuations, and predicting the path of the fire using simulation.

The National Key Centre for Social Applications of GIS at the University of Adelaide is developing such systems for emergency services decision support [27]. We are working with them on addressing the complex distributed computing aspects of this application, including the access to real-time data feeds from multiple sources, and the use of high-performance compute servers to provide rapid simulation.

6 Conclusions

We are investigating the computer systems and software issues related to the efficient storage and dissemination of data in large distributed online archives of geospatial data such as satellite images. We are developing "active" digital libraries, which enable remote data processing as well as remote data access. Technologies such as the World Wide Web, Java, and CORBA are making it much easier to develop portable distributed client/server systems of this kind.

We are currently developing the infrastructure to support a Web-based distributed GIS. Some prototype systems have been implemented, such as the ERIC active digital library for GMS-5 satellite images, and the Imagery Management and Dissemination (IMAD) system for handling defence photo-reconnaissance images. These distributed geographic information systems, and the middleware infrastructure that supports them, are currently being provided with additional functionality and robustness. As the systems are improved, they will be used for advanced decision support applications such as land management and crop yield prediction, and C3I applications for emergency services and defence.

Client/server computing, particularly over wide-area networks, is not yet widely used for GIS applications and research, but we believe the use of distributed geographic information systems that can remotely access and process a variety of data from multiple online data archives has great promise in many research and decision support applications.

Acknowledgements

This work was supported by the Research Data Networks (RDN) Cooperative Research Centre (CRC) and the Advanced Computational Systems (ACSys) CRC, established under the Australian Government's CRC Program.

References

[1] K.A. Hawick *et al.*, DISCWorld: An Integrated Data Environment for Distributed High-Performance Computing, Proc. 5th IDEA Workshop, Fremantle, February 1998.

[2] K.A. Hawick *et al.*, Geographic Information Systems Applications on an ATM-Based Distributed High Performance Computing System, Proc. HPCN, Vienna, Austria, 1997.

[3] K.A. Hawick and H.A. James, Distributed High-Performance Computation for Remote Sensing, Proc. Supercomputing '97, San Jose, November 1997.

[4] K.A. Hawick and H.A. James, Eric: A User and Applications Interface to a Distributed Satellite Data Repository, DHPC Technical Report DHPC-008.

[5] K.A. Hawick and H.A. James, A Web-based Interface for On-Demand Processing of Satellite Imagery Archives, Proc. Australian Computer Science Conference (ACSC) '98, Perth, February 1998.

[6] Roy Williams and Bruce Sears, A High-Performance Active Digital Library, August 1997, http://www.cacr.caltech.edu/ ~roy/papers/sarcan.pdf.

[7] On-Line Data Archives Program of the Advanced Computational Systems Cooperative Research Centre, http://www.olda.dhpc.adelaide.edu.au/.

[8] Roy Williams, SARA Project, http://www.cacr.caltech.edu/~roy/sara/.

[9] NASA Goddard Space Flight Center, Earth Observing System Data and Information System, http://spsosun.gsfc.nasa.gov/New_EOSDIS.html.

[10] European Space Agency, Earthnet Online, ESA Multimission Remote Sensing Product Catalogue, http://gds.esrin.esa.it/reg/catalogue.html.

[11] Australian Centre for Remote Sensing, ACRES Digital Catalogue, http://acs.auslig.gov.au/intro.html.

[12] Shishir Gundavaram, CGI Scripting on the World Wide Web, O'Reilly & Associates, 1996.

[13] Sun Microsystems Inc, The Java Platform, http://java.sun.com/aboutJava/index.html.

[14] The Object Management Group, Common Object Request Broker Architecture (CORBA), http://www.omg.org/corba/.

[15] T.J. Mowbray and R. Zahavi, The Essential CORBA: Systems Integration Using Distributed Objects, Wiley, 1995.

[16] The U.S. National Imagery and Mapping Association, USIGS Geospatial and Imagery Access Services (GIAS) Specification, version 3.1, N0101-B, 5 February 1998, http://www.nima.mil/aig/products/uip/gias/020598/index.html.

[17] The U.S. National Imagery and Mapping Association, U.S. Imagery and Geospatial Information System (USIGS) Architecture Framework, http://www.nima.mil/aig/products/uaf/html/index.html, 18 November 1997.

[18] The U.S. National Imagery and Mapping Association, U.S. Imagery and Geospatial Information System (USIGS) Technical Architecture, http://www.nima.mil/aig/products/uta/html/index.html, 6 November 1997.

[19] U.S. National Imagery and Mapping Association, http://www.nima.mil/.

[20] P.D. Coddington, K.A. Hawick, K.E. Kerry, J.A. Mathew, A.J. Silis, D.L. Webb, P.J. Whitbread, C.G. Irving, M.W. Grigg, R. Jana, K. Tang, Implementation of a Geospatial Imagery Digital Library using Java and CORBA, Technical report DHPC-047, http://www.dhpc.adelaide.edu.au/ reports/047/abs-047.html.

[21] Dublin Core Working Group, Dublin Core Metadata, http://purl.org/metadata/dublin_core/.

[22] Open GIS Consortium, http://www.opengis.org/.

[23] ISO Technical Committee 211 on Geographic Information/Geomatics, http://www.statkart.no/isotc211/.

[24] The Federal Geographic Data Committee, The National Spatial Data Infrastructure, http://www.fgdc.gov/NSDI/Nsdi.html.

[25] The Australia New Zealand Land Information Council (ANZLIC), Core Metadata Elements for Land and Geographic Directories in Australia and New Zealand, http://www.anzlic.org.au/metaelem.htm.

[26] Netscape Communications Corporation, CORBA/IIOP Developer Central, http://developer.netscape.com/ tech/corba/index.html.

[27] National Key Centre for the Social Applications of Geographical Information Systems, Emergency Service System Response, http://ch1.gisca.adelaide.edu.au/ kra/emsr.html.