FUTURE PLANS FOR McIDAS WORKSTATIONS

J. T. Young and Robert J. Fox
University of Wisconsin
Madison, Wisconsin, USA

Abstract

A new architecture for McIDAS is under development. This development work will lead to an open systems based distributed architecture. We envision a significantly enlarged role for McIDAS-X workstations in the near future, to the point where they will be capable of acquiring meteorological satellite and conventional data, building and managing the resulting data base, and functioning as servers to other clients. In this new architecture, MVS and OS/2 based components will be optional. In conjunction with this effort, the McIDAS user interface is undergoing significant revisions and enhancement. The new architecture, its philosophy and status will be described.

1. INTRODUCTION

The McIDAS system was first implemented in 1972 (Suomi,1972) by the University of Wisconsin - Madison (UW). It has evolved continuously over the intervening 21 years. It was originally developed as an integrated system of tools to provide the researcher with a wide range of data access and analysis capabilities (Whittaker, 1977). The system development has been funded by hundreds of research and operationally oriented projects over the years. Each project has expanded the tool kit in directions important to that project. Many of the tools have been rebuilt one or more times as we developed a better understanding of the community's needs, and as technology allowed increased capability with tool redesign. Most of the tools have been developed in an iterative development cycle that includes the user, so that McIDAS is very much a user driven system.

The two major attributes of McIDAS that have fluctuated in this evolution have been the operating system and the centralized versus distributed data base paradigm. The system started in a Datacraft (later Harris Computer) Disk Management System (DMS) with a centralized data base paradigm. In 1978, a distributed data base paradigm was implemented over the Harris DMS operating system. In 1976, the system was first ported to the IBM OS/VS1 operating system, and again in 1983, a full port to MVS based again on the centralized data base paradigm was completed (Suomi, 1983). 1985 saw a return to a partially distributed McIDAS (Rueden, 1985) as the "McIDAS workstations" were upgraded with microprocessors. Now the University of Wisconsin - Madison (UW) is in the process of porting McIDAS to its sixth operating environment, Unix, and converting back to a fully and widely distributed data base paradigm (Santek, 1991).

The port to Unix and the conversion of McIDAS from a centralized system to a distributed system require that the various functions of McIDAS; data base build, science applications, product generation, product distribution and visualization, be redeveloped. This port also provides the opportunity to embrace a number of important standards (X-Windows systems, POSIX, MOTIF, TCP/IP) and a variety of standard data formats.

The move to Unix and specifically to X-Windows systems also affords us the opportunity to implement a new user interface more in keeping with the user interfaces to which the community has become accustomed.

2. THE NEW McIDAS PARADIGM

The new McIDAS paradigm is designed to provide building blocks of functionality, in that each of the functions listed above are modular and can be selectively implemented in response to the user's requirements. There are three phases in the transition to the new paradigm. The previously supported Phase 1 model (figure 1.) required that all data bases be built in a mainframe computer and accessed by workstations locally or remotely attached to it. The mainframe was responsible for the data base build, some science applications, and most of the product generation and distribution functions. The workstation provided most of the science applications and the visualization functions.

In phase 2 (figure 2.) the applications access the data bases using the previous methods, using NFS as client server, or a new client server approach we've called Distributed Data Environment (DDE) described below. This phase is required until all the functions performed by the mainframe are recapitulated in the Unix workstation environment.

Phase 3 (figure 3.) will provide a fully and widely distributed data environment in which an application may access several data bases via Internet during a single execution, extracting data elements from portions of files stored in various formats.

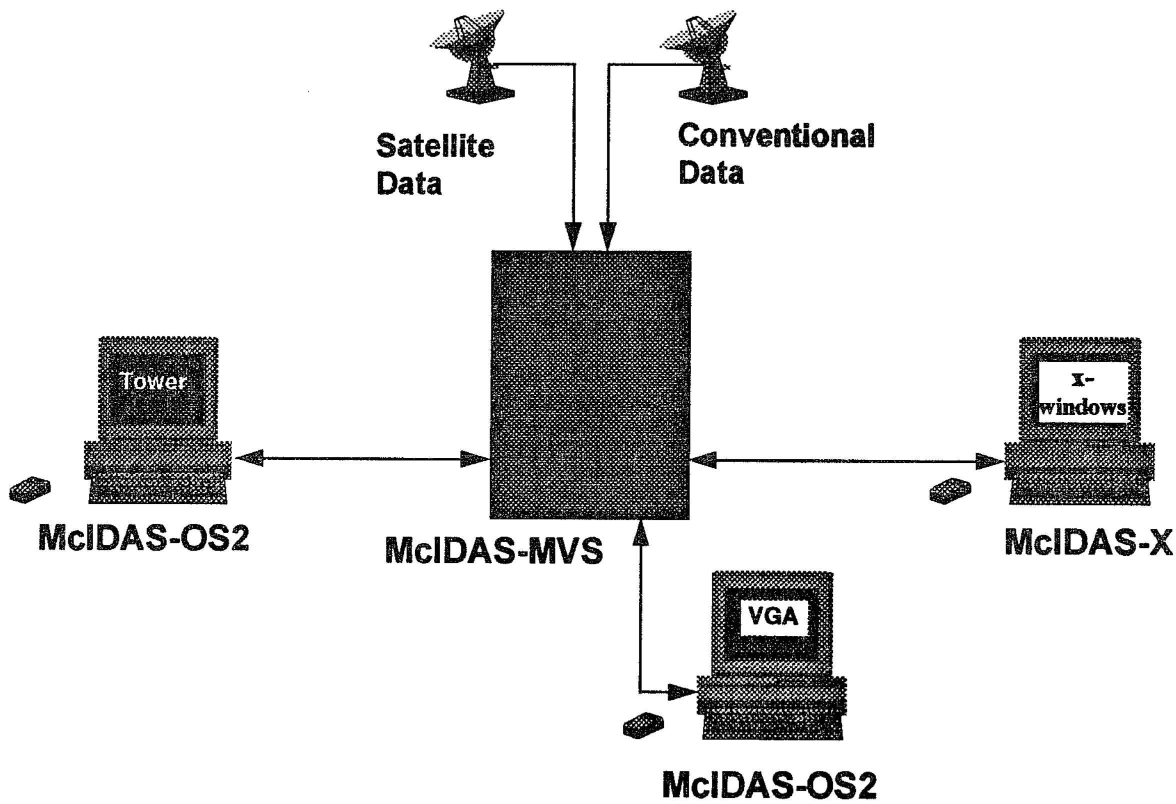

Figure 1. Phase 1 - Previous McIDAS Paradigm

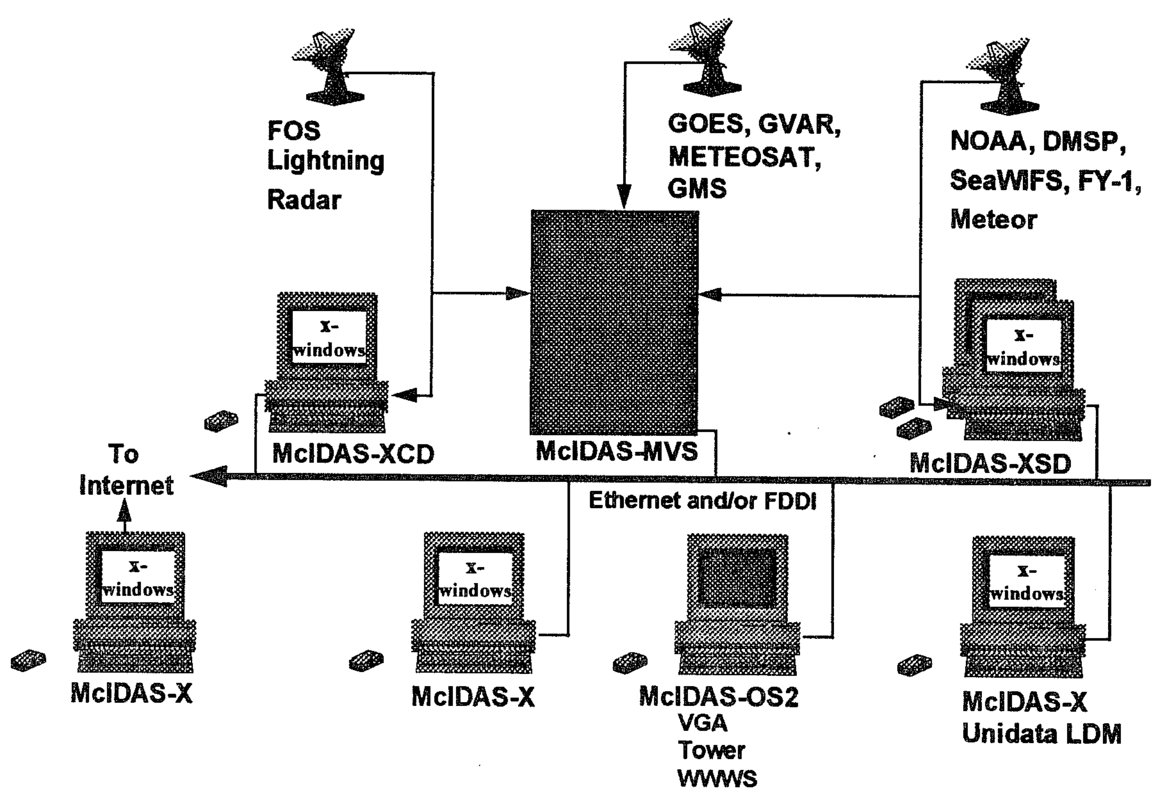

Figure 2. Phase 2 - Partially Distributed Data Environment

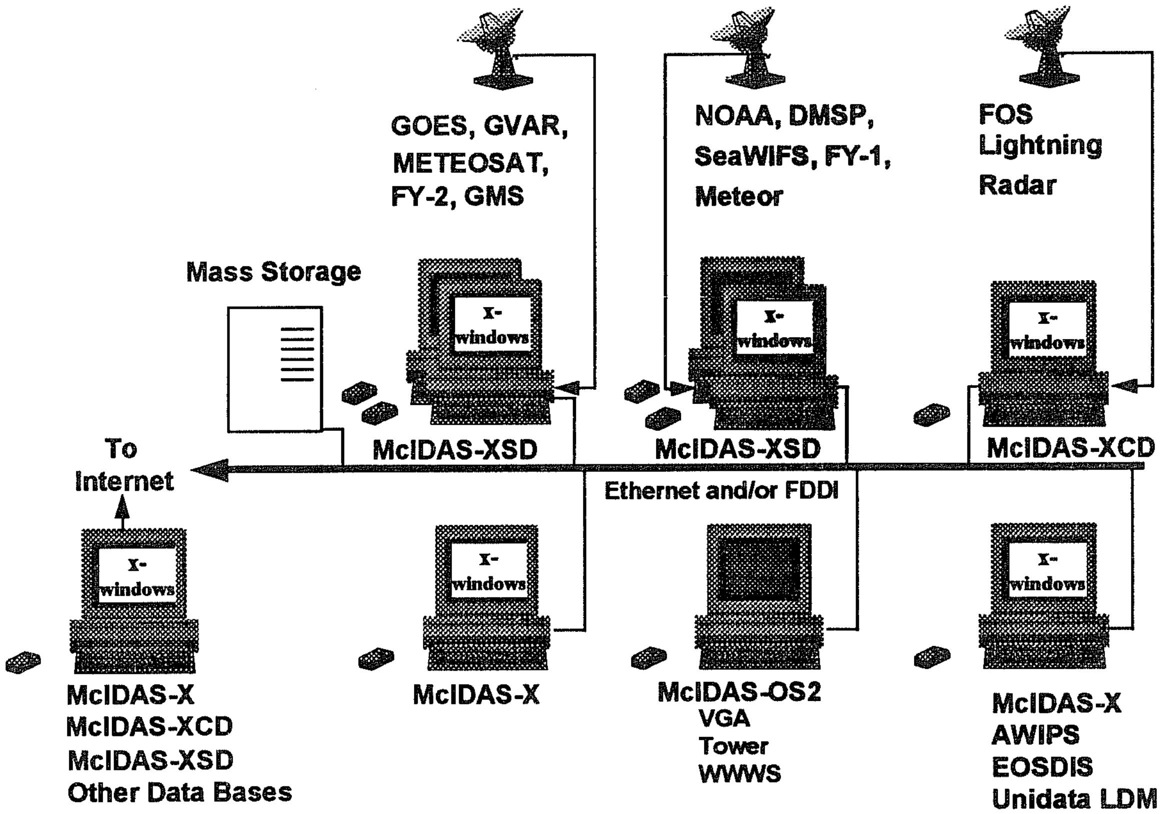

Figure 3. Phase 3 - Fully and Widely Distributed Data Environment

2.1. Distributed Data Environment (DDE)

DDE provides a standard data access methodology for applications accessing data distributed on a network. The DDE has five components: the Applications Standard Program Interface, the client subroutine, the named location table, the TCP/IP socket interface and the server software.

Under DDE, sets of related data files are given group names. The named data groups are usually a time series of data files. Applications use the API to request data of the client (which is a subroutine of the application). The API is intended to provide a sufficient level of abstract so that the applications will be completely independent of the data file format (GRIB, BUFR, netCDF, CDF, HDF, McIDAS, etc.). The API call provides for DDE routing information, data sorting, in addition to information on the return data format such as units, scale, missing value, meta data, etc. The data descriptor is defined in a data dictionary that all DDE applications must use. The TCP/IP location of the server for the named data groups is found by the client in a named location table. The client then sends a request to the server using the special socket interface. The server resolves the data descriptor(s) into the appropriate identifier for the specific format it serves and responds by returning the requested data in the form requested by the application. The conceptual schematic diagram of the DDE concept is shown in figure 4.

Using this approach to data access means that the choice of data formats can be based on other issues, such as transmission efficiency, storage efficiency, etc..

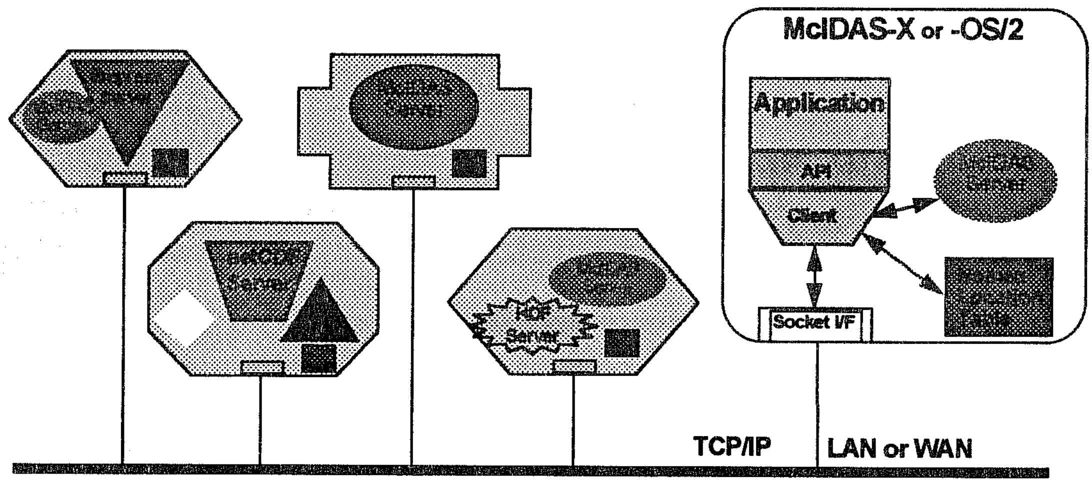

Figure 4. Schematic DDE concept

2.2. Satellite-data Processing Subsystem (SPS)

With transition of the ingestors from the mainframe McIDAS-MVS to Unix, the same opportunity to redesign our approach has been exercised. The next generation of satellite data ingestors separates the satellite unique signal format and data acquisition functions from higher level functions such as ingest scheduling, navigation, calibration and data quality assessment. These functions are physically split between two subsystems; the Satellite-data Acquisition Subsystem (SAS) and the Satellite-data Ingest Subsystem (SIS), respectively. These two systems work together through an interface protocol defined in the Satellite-data Acquisition Interface Specification (SAIS).

The SAS is provided by hardware vendors, and may include all of the components needed to acquire a satellite signal and format the signal contents into sensor data. The SAS may include an antenna, antenna controller, preamplifier, receiver, bit and frame synchronizers in addition to the data formatter and protocol software. The protocol program is the software implementation of the SAIS. That software has been developed at UW and is available as freeware. The SAS is a slave to the SIS, receiving instructions through the SAIS.

The SIS receives the formatted data from the SAS, extracts appropriate meta data from the instream documentation, and files the calibration, navigation and data quality information. The SIS activates product generation software. The SIS is also responsible for generating the antenna pointing information and passing that information to the SAS when antenna control is needed. Normally the SIS resides in an McIDAS-X environment as shown in figure 5. below and the same processor acts as a DDE server.

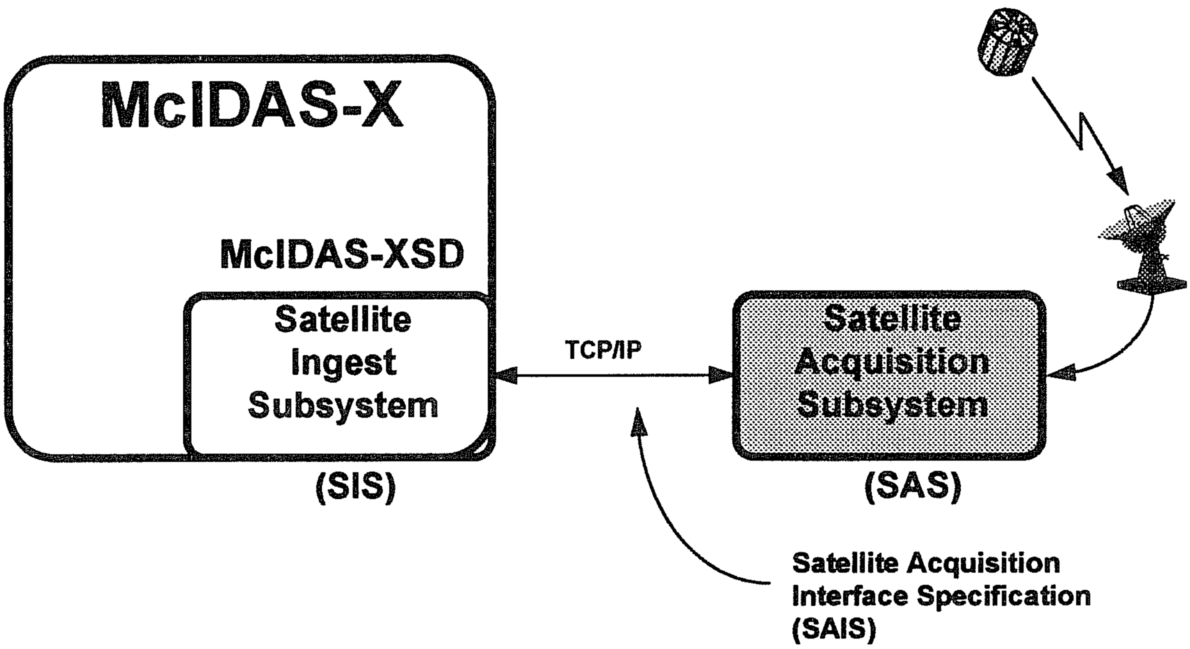

Figure 5. Satellite-data Processing System

3. USER INTERFACE

The McIDAS user interface has been both loved and hated by various constituents in the community. It has been noted for its fast response, freedom of access, and simplicity. It has been criticized as too unstructured, hard to learn, and difficult to understand its capabilities. We are addressing these criticisms with two new graphical user interfaces (GUI). First, control of the display windows has been implemented using a combination of Motif widgets such as VCR control buttons for the loop animation. Second, using a public domain user interface building tool kit, we are in the process of implementing a GUI over the McIDAS command language. The implementation uses widgets and default values to simplify user control of the McIDAS tools.

X-Windows provides services for the user interface that have become a standard for most users. The previous versions of McIDAS have taken advantage of similar resources in OS/2 so as to maintain backward compatibility with prior user interfaces. Several user interface building tool kits are available that facilitate graphical user interfaces on top of a command line interface. We have chosen the Tool Command Language (Tcl, pronounced "tickle") and the Tool Kit (Tk) as the UI building tool kit. It provides a system for developing and using graphical user interface applications. It is a simple scripting language that supports generic programming facilities, such as internal variables, loop and logical flow constructs, and mathematical operations.

In a more exploratory effort, an object oriented approach to a McIDAS user interface has been started. This effort will evaluate the impact of a "drag and drop" methodology combined with natural object paradigms for processes and procedures typical to McIDAS users.

4. CONCLUSION

The system described above provides a modular, easily expandable architecture which will be viable for the next decade in environment data collection and processing. We have targeted January 1995 as the goal for the first implementation of a fully distributed McIDAS-X based system. Several intermediate releases are planned as the various components become available. We have developed detailed plans for DDE, the SPS, and the Tcl user interface. A DDE demonstration package for McIDAS-OS2 has been distributed to interested users, and the results have been highly encouraging. The first SPS is scheduled for installation in March 1994. By July 1994, SPS support for NOAA, DMSP, GOES, and GMS will have been fielded. A prototype operational distributed system will be expanded over the next 9 months, so that support procedures can be developed. The Tcl based UI beta release is scheduled for January 1994 and the beta release of the object oriented UI is anticipated for Fall 1994.

5. REFERENCES

Rueden, J.P., 1985:DESIGNING AN INTERACTIVE SYSTEM FOR TOMORROW'S NEEDS. International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (Preprints of Papers), Jan 7-11 1985, 139-140.

Santek, D.A., T.M. Whittaker, J.T. Young, W.L. Hibbard, 1991: The Implementation Plan for McIDAS-AIX. Seventh International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (Preprints of Papers), Jan 14-18 1991, 177-179.

Suomi, Verner E., November 1972: McIDAS, An Interim Report on the Development of the MAN-COMPUTER INTERACTIVE DATA ACCESS SYSTEM. An internal document of the University of Wisconsin Space Science and Engineering Center, Madison, Wisconsin. Available on Request.

Suomi, V.E., R. Fox, S.S. Limaye, W.L. Smith, 1983: McIDAS III: A MODERN INTERACTIVE DATA ACCESS AND ANALYSIS SYSTEM. Journal of Climate and Applied Meteorology, 22 May 1983, 766-778.

Whittaker, T. M., 1977: Processing and Display of Satellite and Conventional Meteorological Data for the Classroom. Proceedings of a Workshop on Interactive Video Displays for Atmospheric Studies, University of Wisconsin, June 1977.