7.3 SOFTWARE DEVELOPMENT OF THE NEXRAD OPEN SYSTEMS RADAR PRODUCT GENERATOR (ORPG)

Michael Jain[1], Zhongqi Jing[1,2], Hoyt Burcham[1,2], Arlis Dodson[1,2], Eddie Forren[1,2], Jeff Horn[1,2], David Priegnitz[1,2], Steve Smith[3], and John Thompson[1,2]

[1]NOAA/ERL/National Severe Storms Laboratory, Norman, OK
[2]Cooperative Institute for Mesoscale Meteorological Studies, University of Oklahoma, Norman, OK
[3]NOAA/NWS/WSR-88D Operational Support Facility, Norman, OK

1. INTRODUCTION

The NEXRAD Product Improvement (NPI) Program is sponsoring the development of a new Open Systems Radar Product Generator (ORPG) to replace the existing WSR-88D RPG. The NPI is a component of the National Weather Service (NWS) Modernization Plan and is under the purview of the Tri-Agency's (Department Of Commerce, Department Of Defense, and Department Of Transportation) NEXRAD Program Management Committee (PMC). Discussions of the NPI can be found in Saffle et al. (1997, 1998). This paper discusses issues associated with Open Systems and the benefits provided by the ORPG.

2. DISCUSSION OF OPEN SYSTEMS

The following is the IEEE POSIX (Portable Operating System Interface) Committee definition of an Open System:

A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered applications software: (a) to be ported with minimal changes across a wide range of systems, (b) to interoperate with other applications on local and remote systems, and (c) to interact with users in a style that facilitates user portability. [IEEE P1003.0/D15]

2.1 Benefits of Open Systems

The development and marketing of open systems technology has grown considerably in the past decade. In recent years, this growth, coupled with improved networking capabilities, has lead to the implementation of more distributed and client-server type systems that take advantage of open system compliant hardware components. Systems such as these offer many benefits over proprietary or closed systems, including the following:

- A wide selection of multi-vendor, compatible components such as computers, communications servers, and operating systems. The availability of a variety of products provide the system integrators greater freedom in selecting the most effective components available in terms of functionality, performance, cost, and vendor support.
- Supports interoperability across heterogeneous platforms. The computer hardware industry is faced with one of the most rapidly evolving technologies of recent times. For systems being implemented today, it is unreasonable to expect the initially installed hardware (model lines or even vendors) to be available later in the lifecycle when upgrades to the system are required. Long-term strategies of modern systems should include logistics, maintenance, and procurement plans to include a heterogeneous mix of computing hardware platforms.
- Allows upgrading or augmenting the system in a modular fashion as functionality or performance requirements mandate. As funding becomes available and requirements dictate, the latest technology can be incrementally integrated into the system.
- The commitment of commercial suppliers to Open Systems assures future hardware components will be available that can be integrated with existing system components.
- Application software developed on one system will run on another system with minimal adaptation.

These benefits contribute to decrease system dependencies on individual component suppliers and also contribute toward avoiding the need to replace an entire system, as we are faced with today with the current WSR-88D RPG. Software engineered to take advantage of Open Systems compliant hardware components will result in a powerful, economical, flexible, and long-lived system.

2.2 Challenges

Open System technology is not without its own set of challenges. Rapid growth and strong competition in the commercial market can make standardization of like-products difficult to coordinate. Other issues such as incomplete adherence to standards and variations in

Corresponding author address: Michael Jain, National Severe Storms Laboratory, 1313 Halley Circle, Norman, OK 73069-8480; e-mail <Michael.Jain@noaa.gov>

implementation may result in subtle differences between like-products. Accurately assessing the degree of compliance can be difficult. If variations from the standards are detected, adjustments in the system's software would be required when these components are integrated into the system.

Commercially produced Open Systems products are usually developed in a functionally generic fashion to broaden their market appeal. This can add complexity and, in some instances, greater short-term expense of a component over one designed and developed for a specific function.

Customers are also reliant upon vendors for new firmware/software upgrades and any defect corrections that may be required. The ORPG Project, for example, has already been exposed to this issue with defects being detected in vendor supplied software communication libraries. Project team members were required to consult with the vendors and rely on them to correct and deliver the modified software. Willingness of a vendor to cooperate and work with customers should certainly be a criteria for the selection of components.

An important issue to realize is Open Systems does not imply "plug-and-play" hardware or software. There will usually be some level of effort required to integrate a new hardware component into the system. Even for well engineered and open system compliant software, some porting or new software development efforts should be expected to successfully incorporate other "Open Systems" hardware components or operating systems. However, due to the defining nature of Open Systems, the efforts of commercial suppliers to abide to standards, and the wide selection of products available on the market, integration efforts can be minimized by selecting the proper components.

Even with these challenges, the benefits of Open System technology far outweigh the detriments. The flexibility and opportunities available from Open Systems technology are simply not available in proprietary systems.

3. OPEN SYSTEMS AND THE ORPG

The ORPG is being designed such that the resultant software is portable, reliable, expandable, scalable, and maintainable (Jain et al., 1997 and Jain, 1997). These concepts are the embodiment of Open System technology. While previous papers (Jain et al., 1997 and Jain, 1997) have discussed the software architecture of the ORPG, the current paper will concentrate on some of the implementation aspects of the ORPG.

3.1 Portability

Software portability is a fundamental concept of Open Systems and has been established as a firm system requirement of the ORPG. To satisfy this requirement, a layered software architecture (Jain ,1997 and Jain, et al., 1997) as well as an adherence to open systems standards are being utilized in the development of the ORPG.

A layered software architecture model contributes to isolating the application level programs from the underlying system infrastructure and operating system dependent layers. If the need arises to port the software to another operating system, only the software modules in the operating system dependent layer will need to altered, thus shielding the application layer of the ORPG.

Adhering to Open System standards includes using a compliant software development environment. For this reason, the ORPG is being developed in the C programming language and under the UNIX operating system environment.

ANSI C is a well established, standardized programming language quite suitable for the development of the ORPG. Consideration was given to the C++ programming language. However, at the time of project initiation, standardization of this language had not been established leading to concerns about portability. The modularity of the ORPG software will allow for the use of other programming languages, such as C++, to be introduced to the system as deemed appropriate.

UNIX is a well established, POSIX compliant operating system. Several commercial vendors currently support UNIX and there are also public domain versions of the operating system available. Development of the ORPG software is currently being done using the HP-UX and Solaris operating systems for workstation platforms. Recent work has also been started to include development on a Personal Computer (PC) hardware platform using Solaris x86. Solaris x86 is Sun's version of the Solaris operating system designed for the Intel microprocessor used in the personal computer market.

3.2 Distributed Processing

The ORPG is being designed to be a distributed processing system functioning as a networked system of computing workstations, communications hardware, archive devices and the like. This architecture is well suited for taking advantage of the wide array of Open System hardware components that are commercially available and has several other advantages as well.

3.2.1 Increased Computing Capacity

As new meteorological algorithms and products are developed by the research community and transfered to operations, additional computing resources will eventually be required. Instead of having to replace the saturated workstation with a machine of greater capacity, the system can be augmented by adding workstations to the ORPG network to satisfy computing resource requirements. Furthermore, due to the rapid evolution of the computing hardware market, the system is being engineered to accommodate a heterogeneous mix of hardware components should that be required.

Another approach to enhance the computing capacity of the ORPG is through multi-processing, or the use of two or more processors in a single workstation. As long as the operating system supports multi-processing, no changes in the ORPG software would be required. This approach can be used to increase computing capacity. However, this alone does not address the added benefit of redundancy and increased system reliability that is garnered by the distributed processing approach.

3.2.2 Fault-Tolerance

Multiple networked workstations, coupled with software engineered to support dynamic reallocation of processes and data stores, provides fault tolerance in the areas of load-leveling and the catastrophic failure of hardware components.

3.2.2.1 Load-Leveling

Specialized processes designed to monitor the health and well-being of a multiple-workstation ORPG will detect situations when a particular workstation's computing resources become saturated. Once this situation is detected, the system will determine what processes should be reallocated from the saturated workstation to one being less utilized. This will help prevent load-shedding episodes due to various resource exhaustion and improve the efficiency, reliability, and responsiveness of the system.

3.2.2.2 Hardware Failure

With a multi-workstation ORPG, specialized processes monitoring the ORPG system will detect the loss of a workstation due to hardware or network failure. The ORPG system will have knowledge as to what processes and data stores were running or stored on the failed workstation. These processes and data stores will be automatically reallocated to the surviving computing workstations on the ORPG network. In most cases, this event should occur with no or minimal loss of data or function.

3.2.2.3 Software Failures

A significant advantage of the ORPG will be its resilience to software task failures. The failure of any ORPG software application should not adversely affect any other application or cause the ORPG to fail as a system. The distributed environment with applications running as independent processes and indirect interprocess communications via linear buffers (Jain, 1997) act to contain failures of individual tasks. Further, the operating system serves to partition memory of the independent tasks preventing memory corruption of other tasks. In the event a task fails, the individual task can be restarted with little or no impact on the rest of the system.

3.3 Communications

The ORPG is required to support the communication protocols and interfaces of the existing WSR-88D such that the transition to the ORPG is transparent to current end users. The ORPG design is also required to support the evolution of communications to new and improved protocols and interfaces. These requirements, coupled with the issues described earlier in using open systems hardware places a challenge in the design and implementation of the communication subsystem. A discussion of the ORPG communications subsystem can be found in Jing, et al. (1998).

The legacy communication protocols include X.25/PVC for the narrowband and HDLC/ABM for the wideband. Evolutionary protocols and interfaces may include TCP/IP. A problem encountered with the legacy protocol is that while the X.25 family is a standard communication protocol, a standardized interface does not exist. Because of this, the Application Programming Interface (API) developed and supplied by various vendors will be quite different from one another. In contrast, "evolutionary" protocols, such as TCP/IP, have a standardized interface that can be expected to be consistent from one vendor to another.

The ORPG implementation includes a software module that contains all of the hardware dependant code required for a specific piece of communication hardware. This module, referred to as the "comm_manager", supports an established Communication API for the ORPG and follows the vendor supplied API for the communications server. Effectively, the comm_manager establishes a bridge between the ORPG and the communication server, isolating the ORPG from whatever hardware or protocol has been chosen for implementation. A separate comm_manger would need to be developed for each vendor supplied communications server supporting X.25. However, only a single comm_manager would need to be developed to support a standardized communications interface such as TCP/IP. This design allows the system to have the flexibility to utilize a wide variety of commercially available products.

4. IMPROVED HUMAN COMPUTER INTERFACE (HCI)

The Unit Control Position (UCP) for the current WSR-88D is a collection of hardware and software that is usually located in a specific location within a National Weather Service Forecast Office. It is an alphanumeric, nested menu interface that requires a considerable amount of user training and familiarization. Due to an ineffective interface design, several areas of functionality are available but are rarely used.

In contrast, the ORPG HCI is a family of windows that can be opened on any ORPG compatible workstation having a network or serial connection to the ORPG. The ORPG HCI is being designed and implemented as a graphical, intuitive interface that will reduce training, provide effective display of system status information, and improve system control procedures. Priegnitz et al. (1997, 1998) describe the HCI being developed for the ORPG.

One of the design goals of the ORPG HCI is to produce a very intuitive and effective interface for the end user. The graphical characterization of the system uses color, motion, sound, and graphical depictions throughout the HCI to increase the effective transfer of information and control of the WSR-88D system.

The ORPG HCI is also being designed and implemented in a fashion to allow multiple instances to be run on several workstations simultaneously for the purpose of displaying system status information. This provides status information to a greater number of people within the user environment. Authentication and lockout controls will be imposed on HCI system control functions to assure only authorized users are allowed control access and to assure only one user at a time is allowed to issue specific control commands to the radar.

5. ACKNOWLEDGMENTS

The authors would like to acknowledge the Office of Systems Development of the National Weather Service for funding this project. We would also like to thank members of the WSR-88D Operational Support Facility Open Systems Team that have participated in this development project.

6. REFERENCES

Jain, M.H., 1997: Software architecture of the NEXRAD open systems radar product generator (ORPG). Preprints 28th Conference on Radar Meteorology, Austin, TX, AMS, Paper P3.1.

Jain, M.H., Z. Jing, A. Zahrai, A. Dodson, H. Burcham, D. Priegnitz, and S. Smith, 1997: Software architecture of the NEXRAD open systems radar product generator (ORPG). Preprints 13th IIPS, Long Beach, CA, AMS, Paper 8.4.

Jing, J., M. Jain, H. Burcham, 1998: User profile based product distribution in the WSR-88D open systems radar product generator (ORPG) , Preprints 14th IIPS, Phoenix, AZ, AMS, Paper 7.4.

Priegnitz, D.L., E. Forren, and M. Jain, 1998: Update on the human computer interface for the WSR-88D (NEXRAD) open systems radar product generator (ORPG), Preprints 14th IIPS, Phoenix, AZ, AMS, Paper 7.5.

Priegnitz, D.L., D. Frashier, T. Marci, T. O'Bannon, S. Smith, R. Steadham, M. Bullard, and M. Rausch, 1997: Human factors methods in the design of the graphical user interface for the open systems radar product generation (ORPG) component of the WSR-88D. Preprints 13th IIPS, Long Beach, CA, AMS, Paper 8.8.

Saffle, R., and L. Johnson, 1998: NEXRAD Product Improvement: An Overview of the Continuing Program to Evolve the WSR-88D System to an Open Systems Architecture, Preprints 14th IIPS, Phoenix, AZ, AMS, Paper 7.1

Saffle, R.E. and L.D. Johnson: NEXRAD Product Improvement Overview, Preprints 13th IIPS, Long Beach, CA, AMS, paper 8.1.