BENEFITS, PROBLEMS, AND ISSUES IN OPEN SYSTEMS ARCHITECTURES

A report prepared for the IEEE Task Force on Open Systems, P. Emmerich and J. Britton, Co-Chairmen. Contributing authors: P. Emmerich, P. J. Traynor, S. A. Klein, M. T. Fisher, R. D. Burn, R. Hoffman, and G. Castelli

ABSTRACT:

This paper is sponsored by the Power System Control Centers Working Group and the Control Centers Open Systems Task Force. The intent of the paper is to focus industry attention on issues relating to open energy management systems (EMS).

The short note papers address both benefits and problem areas relating to the current open systems environment. The issues considered herein should be considered by any utility planning for a new EMS. They should also stimulate further thought on related topics that will ultimately affect EMS reliability and availability.

INTRODUCTION

Paul Emmerich, Member, IEEE
Salt River Project

One of the major concerns of any utility considering the procurement of a new EMS is the impact that the open systems revolution will have on the planned EMS. Past generation EMS have been easily maintainable without major software or hardware changes for periods ranging from five to 15 years. This was possible because of the relative stability of hardware and operating system software platforms supporting the specialized EMS applications. Typically, such large numbers of these platforms were sold that assistance in solving problems could be obtained for years after EMS installation, even if the original vendors were no longer in business. Such stability is no longer possible. Basic hardware platforms are changing almost yearly. Key operating system standards such as real time POSIX are not yet readily available. Hardware components are no longer maintainable at the component level by utility personnel. Availability of spare boards and vendor assistance for hardware that has not been upgraded for five years is questionable at best. Because of the use of third party packages by most EMS vendors today, projecting the availability of support to meet the needs of a 24 hour a day operation has become more complex as the stability of multiple vendors must be considered. Use of third party software where source and internal documentation may not be available raises questions of long-term interoperability.

On the other hand open systems technology offers the promise of many benefits. Applications that support end user needs are much easier to enhance to meet changing requirements. Both hardware and software costs are considerably lower for the initial purchase. Systems can be upgraded with more powerful hardware when needed without requiring major software rework in many cases. These and many other advantages make open systems very attractive for meeting end-user requirements.

This collection of notes addresses topics that should be considered by utilities both for planning a new EMS procurement and for considering the need to get involved in the standards development process. As has been demonstrated by the history of OSI, users of new technology must take an active interest in the emerging standards and make their needs known to those involved in the standards development process. Where the general standards will not meet the needs of a particular special interest group, that group must develop adjunct standards that build on the base package.

The authors of this paper hope that the topics considered herein will stimulate thinking on how best to take advantage of the open systems benefits while still meeting the high availability standards that EMS users rightly demand.

MAINTENANCE OF AN "OPEN" ENERGY MANAGEMENT SYSTEM FROM A UTILITY PERSPECTIVE

P. J. Traynor, Member, IEEE
Wisconsin Power and Light Co.

With the benefits of an "open" Energy Management System come a number of challenges that the EMS engineers have to face. These challenges include increased software maintenance costs, increased capital expenditures, integration difficulties, and increased documentation and training.

At Wisconsin Power and Light we have continually changed and added to our Energy Management System since it went in service in 1978. Incremental upgrades, including new unit commitment, transaction scheduling system to handle wheeling, and network analysis functions, have allowed us to continually enhance the functionality provided to the system operators. The closed nature of the computer system we were modifying presented us with many difficult challenges. Many of the experiences we had are also applicable to anyone maintaining a more open Energy Management System today. I will discuss each of the challenges we faced in the following paragraphs.

Increased Maintenance Cost

New application software may require the latest revisions of supporting software and increase annual software maintenance fees. For each piece of software licensed, there is a 10-12% annual maintenance fee based on the license cost. Therefore, whenever new software is licensed, its effect on your maintenance budget must be considered. It is important to periodically assess whether each product is meeting your needs and worth the annual cost. Sometimes the annual maintenance cost can be reduced by moving existing software to new platforms, such as workstations. This can reduce maintenance costs while maintaining existing functionality. We moved our relational database from a mini-computer to a workstation server and saved approximately $20,000.00 per year.

In the past, we licensed the computer operating system and a limited number of products for which we paid an annual maintenance fee. By adding functionality, we now pay maintenance fees for the operating system, compilers, networking software, relational database, relational database access software, document handling software, spreadsheet, and other software. Adding or upgrading

application software often requires the current version of the operating system or other support software.

Sometimes additional hardware is required to accommodate the new software or functionality and that hardware needs to be maintained.

Integration Difficulties

Even with a very flexible networked computer system, there are still integration difficulties. Keyboard incompatibilities require new keyboard software to be written; sometimes software works fine for one user, but for some reason another network user is unable to access it. There are limitations with emulation software and situations where one device can access other computers on the network, but none of the other computers can access that device. There are also some difficulties because vendors use different versions of popular standards, such as X-windows and Motif. This makes it difficult for utilities to have a consistent user interface.

Increased Annual Capital Expenditures

With an Energy Management System that is more flexible, we can meet the system operators' needs more effectively. This requires yearly capital expenditures to make the changes to hardware and software to meet new requirements. For example, if the system operators justify the addition of a weather radar receiver system, the receiver must be purchased and installed; and the operators need to be trained. If additional computer networking is required to provide access to data, a capital investment is again needed. Considering the time value of money, these yearly capital expenditures often cost less than a full Energy Management System replacement every fifteen years.

The annual capital expenditures normally require a different budgeting philosophy than a utility may have followed in the past. It could be difficult to have the new budgets approved until the benefits of maintaining an "evergreen" (i.e. always current) system are proven to be cost effective. Continually increasing system functionality instead of periodic system replacement costs less both in procurement dollars and in staffing requirements.

Increased Training and Documentation

Due to the many software products and hardware components it takes longer to train someone to be effective in the Energy Management System engineering area. Especially in small to medium size utilities the engineers must be knowledgeable in current networking and computer technology and power system operations. Most new software is written in "C" and relational databases are utilized. Existing support staff must be trained in these areas.

There is also an ongoing requirement for system operator training based on incremental upgrades, new operator interfaces, and enhanced functionality.

An open Energy Management System requires stricter documentation procedures since the goal is to never do a full scale replacement. Any documentation deficiencies will not go away due to system replacement. It is critical to set up documentation standards and a method to ensure their use. Sometimes third-party documentation must be enhanced in order to satisfy in-house quality and consistency standards. A peer review process is one method that has proven effective in assuring both documentation quality and consistency.

Conclusion

Overall, an open Energy Management System is desirable from a utility perspective. To effectively maintain this type of system, additional money must be budgeted for both annual O & M and capital expenditures. The EMS support staff must be trained to utilize new software and hardware and any programming done within the utility must be appropriately documented. Integration difficulties should be expected and dealt with. An adequate support staff needs to be in place. By doing all this, the EMS support staff is able to better meet system operator needs and extend the life of the Energy Management System.

OPEN SYSTEMS ISSUES: STANDARDS AND TECHNOLOGY
Stanley A. Klein, Member, IEEE
Atlantic Research Corporation

This note addresses some of the standards- and technology-related issues that are encountered in attempting to implement an open systems architecture. The issues addressed include:

- Providers' love-hate relationship with standards
- Importance of user "clout"
- Competing standards and technology
- Marketplace availability of compliant products
- Dependence of usability on underlying focus

Providers' Love-Hate Relationship with Standards

Open systems standards are a mixed blessing from the provider viewpoint. Incentives for compliance include the market opportunity presented by user demand; the relative ease of market entry for standards-compliant products; and the provider infrastructure costs that can be avoided by available approaches to providing standards-compliant products (such as purchasing, porting, and reselling a compliant package).

Disincentives are based on the fact that standard products can easily come to be treated as commodities, where all products are regarded as equal and price competition predominates. For example, IBM-compatible personal computers have become a commodity product -- "no-name clones" are regarded as competitive with brand-named products. It is difficult for a provider to maintain market share (and sometimes even financial stability) in a commodity market.

There are a number of strategies used by providers to mitigate the disincentives. These strategies attempt to preserve a proprietary role while supporting open systems standards. Examples of these strategies include:

- Providing proprietary extensions to the standard features

- Releasing to the public domain selected interface data for proprietary products

- Developing standards through private, multivendor consortia

- Providing inter-vendor interoperability by means of proprietary products

- Using pricing strategies to induce users to choose proprietary products over standard products

- Using market "clout" to either ignore standards or to make proprietary products into ad hoc standards

Examples of Standards with User "Clout" Behind Them

Open systems standards will be implemented in the marketplace only if there are clear benefits to providers and/or insistence by users. The U.S. Government and its individual agencies have mandated several standards for use in new systems. The mandates generally require use of the standards unless there is extensive justification and high level executive approval of waivers. The standards being mandated include the Government Open Systems Interconnection (OSI) Profile (GOSIP), the Structured Query Language (SQL) for relational databases, the Ada language, the Portable Operating System Interface (POSIX), and the Computer-aided Acquisition and Logistics System (CALS, that specifies OSI engineering drawing/document standards not yet included in GOSIP). GOSIP and CALS are close relatives of the Utility Communications Architecture (UCA).

The standards are moving slowly at present, but their use will accelerate as major new systems are brought on line, as existing systems undergo major upgrades, and as needs are recognized for interoperation between existing systems and new/upgraded systems.

Competing Standards and Technology

Rather than being a static collection of well-accepted approaches, open systems standards tend to reflect the results of dynamic competition among evolving and changing technologies. It is critical for the user to remain alert and flexible. Some examples of changes in open systems technologies include:

- The eclipse of the IEEE-696 (S-100) Bus when the IBM PC was brought into the market

- The continued strength of the TCP/IP data communications standards, that are "supposed to be replaced" by OSI

- The appearance of microprocessors with power exceeding that of older mainframes

- The potential competition between Unix and Windows-NT

- The changing focus of database technology from CODASYL (a decade ago) to relational (currently) and possibly to object-oriented (in the future).

Marketplace Availability of Compliant Products

Development and adoption of a standard -- in many cases called a "base" standard -- is only a first step. The product cycle to market availability is often difficult to forecast. Further steps may include preparation of elaborative standards (such as implementers agreements and conformance test standards), development of prototypes and products, and adoption by users. Initial implementations may not provide the full capabilities supported by the standard. Full capabilities can take years to appear in the marketplace.

The X.25 standard provides an example. X.25 was adopted in 1980 and revised in 1984 and 1988. To date, standard conformance tests are available only up to the 1984 version and most common carriers are using either 1980 or 1984.

Dependence of Usability on Underlying Focus

The features of a standard depend significantly on factors such as the applications experience of the people involved in development of the standard, the assumed environment, and the application on which the original concept was focused. The underlying focus of a standard influences tradeoffs and constraints that can create impediments and inefficiencies when use of the standard is extended to other areas.

For example, under the OSI implementers' agreements, the most urgent delivery priority under the X.400 Message Handling System is 45 minutes. This is clearly appropriate for administrative urgency ("I need immediate price quotations on the following items...") and reflects lack of consideration for operational urgency ("The system is operating in emergency state; take the following actions immediately..."). Similarly, the SQL standard was developed to support ad hoc retrieval from business databases; the Unix operating system was developed to support research and academic computing. Any real-time applications of SQL or Unix represent extensions outside the original areas of focus.

Conclusion

The significant benefits of open systems can be achieved only through diligent attention to a variety of risks and issues.

AN OPEN SOLUTION TO THE DEVELOPMENT SYSTEM PROBLEM

M. T. Fisher, Nonmember, and R. D. Burn, Nonmember
ABB Systems Control Company, Inc.

Nearly every EMS/SCADA system project has a requirement for the vendor to supply a development environment to the customer in which database, display, and often software development can occur. A decade or more ago the supply of the hardware and software resources necessary to provide this functionality at the customer site was a significant effort. Due to the size and cost of computers and related peripherals these capabilities could often only be economically provided through the installation of remote batch/interactive terminal equipment. Remote equipment was connected to the vendor site through voice grade communication links, which often gave inefficient performance.

This paper presents as a reference point a case history of the problems encountered in providing a development environment to ABB Systems Control's first complete-system EMS customer. In this case history a more radical approach to the development system was used, with undesirable side effects. A discussion of the support of an open solution to the development problem to a current customer is presented. Finally a vision to the future is mentioned.

American Electric Power EMS

Prior to 1980 ABB Systems Control (under a previous ownership) was a software supplier to other EMS/SCADA vendors. In the 1981-1983 time frame development occurred on the first complete EMS system for the American Electric Power control center under construction in Columbus, Ohio. At that time AEP headquarters were located in Manhattan, New York, where the customer required an environment in which the database, displays, and some applications were to be developed during the course of the project. In this project the cost of a third computer for development purposes was viewed as too costly. The solution to the development problem was the splitting of the redundant hardware configuration between the customer site in Manhattan and the Systems Control facility located in Palo Alto, Calif. Systems Control installed database and display generation software at the customer site and updated that software periodically throughout the project.

Prior to the beginning of commercial operation in the fall of 1984, both halves of the EMS configuration were installed in Columbus, where final systems integration occurred. The exercise of

integrating the failover and database backup software was, by today's standards, a lengthy and painful effort. It was clear to all that this approach should never be repeated on future system projects. By the mid 1980's lower priced members of the same processor family as the main EMS CPUs were available. This allowed the configuration of moderately-priced development systems. In recent cases, the development environment consists of several workstations that will eventually be integrated into the system as part of the User Interface (UI).

Wisconsin Power and Light EMS

In the fall of 1991 ABB Systems Control began work on a large Energy Management System for Wisconsin Power and Light Company's System Operation Center (SOC), located in Cottage Grove, Wisconsin. The configuration consists of a pair of VAX 6000 model 620 host computers that are slated to be connected to full-graphics UI consoles, based on the new Alpha 64-bit RISC architecture from Digital Equipment Corporation. Since the project workstations would not be available until about a year into the project, a solution to the customer development environment was not readily apparent.

The existing WP&L EMS consists of Data General real-time computers, that executed SCADA and AGC, and a collection of Data General Avion UNIX workstations, used for database generation and the execution of WP&L-integrated network applications. Early in the project it was discovered that WP&L had a VAX 6000 model 310 computer in the corporate headquarters in nearby Madison, which ran an application. This VAX was replaced by a low cost VAX 3100 to run the single application, and the VAX 6000 was moved to the SOC and later upgraded to a model 610 through an in-cabinet upgrade. Since the ABB Systems Control full-graphics UI software was based on X-windows and Motif, the ability to use the Avion workstations as X-terminals seemed the ideal solution prior to the availability of the Alpha workstations. With this approach, the X server would run on the Avion, while the X client and the UI application would execute on the VAX 6000.

After solving a relatively short list of problems, the interoperation of the VAX and the Avions proved to be an excellent solution. The following is a list of the specific technical steps that were taken to arrive at an operational windowing solution. These steps are documented in detail to aid others in configuring a multi-vendor graphic application environment.

1. The VMS DECwindows Motif layered product was installed on the VAX 6000 so that the various toolkit routines could be linked into the UI application program image.

2. The VMS ULTRIX Connection layered product was installed on the VAX 6000 to provide TCP/IP as a transport mechanism over which the X protocol could communicate between the Avions and the VAX host.

3. In the context of the VAX process in which the UI application runs, the necessary logical name must be created to connect to the X server on the Avions. This name is created by the VMS DCL command

    SET DISPLAY/CREATE/NODE='node_name'/TRANSPORT=T CP/IP

    where node_name is the network node name of the Avion.

4. The UI application must connect to the X server at run time through the Motif toolkit routine:

XOpenDisplay (dispname);

dispname has the form host:server.screen. In the case where DECnet serves as the transport mechanism, dispname has the form host::server.screen. Since the UI application normally runs in a DECnet environment the DECnet form had to be changed to the TCP/IP form, after which the application was compiled and relinked.

5. The Avion workstations must be configured with font and color files in order for the X server to operate properly. Since the UI application uses standard fonts, no customization of fonts was required. Since the UI application uses a specific set of colors, the color file rgb.txt had to be customized. Basically this text files defines for each named color the percentages of each of the three primary colors that comprise the specified color.

After successful interoperation of the VAX host computer and the Avion workstations, WP&L integrated existing Tektronix X-terminals to supply additional developer positions. Due to the multiple UI processes, the memory consumption on the VAX 6000 increased as additional X-window devices were added to the system. After the experience of the Avion integration, the support of the Tektronix X-terminals proved to be quite simple. It should be pointed out that this solution was possible due to the existence of two de facto standards: the X-window client-server model and protocol, and the Motif windowing system made available by the Open Software Foundation. One annoying problems still remains, however, in that a lack of standards in keyboard layout and interface causes problems in achieving identical functionality across multiple X-window terminals.

Future Vision

The existence of industry standards has permitted the achievement of interoperability that would have been impossible ten years ago. There are, however, annoying problems that need to be resolved before complete ease of interoperation is commonplace.

The experience of the configuration of the WP&L development systems has enabled ABB Systems Control to improve the installation procedures in two ways: better installation documentation and simpler configuration of options. Once the range of supported configurations has been completely tested, the installation documentation can clearly state the various steps to be performed by installation personnel. The specification of the transport mechanism can be performed in configuration files, rather than by code modification.

With the advent of small form-factor workstation products with three-digit Specmark ratings, the ability to provide very inexpensive, but powerful development environments is here. It now costs as much to discuss the options as it does to procure the equipment, with "basic box" prices of under $10,000 for a workstation.

OPEN SYSTEMS: SOME OBSERVATIONS FROM A VENDOR'S PERSPECTIVE

R. Hoffman, Member, IEEE and G. Castelli, Nonmember
CAE Electronics Ltd

Introduction

Our company has for many years had a tradition of adopting industry standards where ever possible. For example, our RTU communication protocol has been using HDLC, a layer 2 function in the EPRI UCA profile, for over 10 years. Our experience with HDLC has been very positive.

In the mid 1980's we recognized the benefits of the Ethernet local area network technology as an alternative to a large amount of point to point wiring in distributed SCADA/EMS system. As we began the necessary development activity one of the first issues to be faced was which LAN protocol to use. Although our host computers were VAX we decided DECNET was unsuitable partly because our front end processors and our MMI consoles were non DEC and equally importantly DECNET did not support dual-redundant LANs. A single LAN would be a single point of failure which is unacceptable in the SCADA/EMS application.

In our desire to follow the latest industry standards we decided to develop our own LAN protocol following the IEEE 802.3 standard. Our implementation was successful and is operational in numerous control centers in several countries. In hindsight from the 1990s the decision was less than optimal since TCP/IP emerged as the de facto Ethernet standard, supported by every workstation manufacturer. Our current product offering supports TCP/IP over IEEE 802.3.

The point is that following the latest international standard does not necessarily equate to the most open solution. Even today, a vendor can try to be avant garde and implement systems based on the EPRI UCA profiles or he can make a system that works with a wide variety of workstation platforms by picking TCP/IP. We feel that EPRI/UCA may have chosen the wrong communication profile (OSI) given the rate at which TCP/IP networks are being deployed. Actually the ideal profile is one that is transparent to the application.

It is perhaps risky for the electric power industry to attempt to develop its own standards. The real time SCADA/EMS market is only a very small percentage of the total computer industry market and it is probably far more important for a given utility to be in line with the general computer industry than with an industry specific profile.

Some Weaknesses of Current Standards

We believe that the current computer industry drive towards open systems and the desire of utilities for systems based on open architecture is appropriate. Our own company has invested heavily in LAN communications facilities for our internal operations. Recently we upgraded our network by installing an FDDI backbone connecting together a number of Ethernet-based subnetworks. Approximately 450 computing nodes are currently supported.

Our current product development efforts are directed towards continually improving the openness of our system. The benefits of an open system are numerous and need not be repeated here.

In some areas such as X-Motif graphical user interface our experience has also been very positive, except perhaps for the very large amount of technical material to be mastered. However, current computer industry standards fall short of meeting all of the traditional SCADA/EMS requirements in several respects, primarily with respect to high availability and system performance.

High Availability

Neither UCA/OSI nor TCP/IP support automatic switch over to an alternate LAN transparently to the application. Both expect the application to do a retry. Our position is that this is unacceptable; vendors need to implement their own solutions to this problem. Similarly the off the shelf client-server remote procedure call

standards such as OSF/DCE do not provide adequate robustness to failures in either end of the client-server pair.

System Response

UCA/OSI does not support a LAN broadcast mode with guaranteed message arrival. Broadcast mode is a highly desirable feature in large distributed SCADA/EMS systems both to meet the RTU to CRT screen data transport times required by most utilities and in order to minimize LAN traffic. The guaranteed message arrival is required to ensure that no alarms are lost - a standard SCADA system requirement.

Reliable High Volume Data Processing

Commercial relational database technology has developed rapidly in recent years and offers capabilities far richer and more flexible than what a typical EMS vendor can afford to develop on his own. Most vendors are including a commercial RDBMS as part of their system. However, in general RDBMS's are unable to support the high rates of data transactions required in SCADA systems and therefore some memory resident database structures are still required. Currently there are no industry standards to manage memory resident databases in distributed systems so we have found it essential to develop our own techniques.

Another issue in the use of commercial relational databases is that while all RDBMS appear very similar, for example, in terms of supporting basic SQL queries, there is considerable variation from product to product in terms of the advanced features supported. In order to obtain the functionality and performance requirements of the EMS, RDBMS product X may technically be clearly superior. However, many utilities have decided at the corporate level to select RDBMS product Y as the company-wide standard. In this case the vendor must choose between offering the technically superior product X or to be "open" in the context of that company with product Y.

The use of memory resident databases as well as RDBMS results in the need for an overall data server which makes the location and type of database transparent to the application programs. This unified access to heterogenous databases is one of the major objectives of the EPRI DAIS project. However, the DAIS requirements are recognized as going beyond what is currently commercially available and vendors must develop their own solutions - or wait until these capabilities become available.

Open Communication Standards

Another area where there is a lack of appropriate industry standards is in radio communications to pole-mounted RTUs. The bulk of the communications standards seem to be inappropriate for radio media in the distribution automation environment where a large number of RTUs must communicate on a single frequency.

We have found it necessary to develop our own communication protocol for radio-based distribution SCADA systems. The EPRI UCA effort neglected to address this important area of communications.

From the point of view of a vendor providing systems in the international arena the lack of standard application protocols for inter-utility data exchange will probably remain a fact of life for many years to come.

Conclusions

While the move to open systems should ultimately benefit everyone involved, it must be recognized that existing closed or

proprietary databases, communication protocols and even software packages will be around for some time to come. For example it is to be expected that proprietary old fashioned RTU protocols must continue to be supported by the new generation of open systems. With this in mind we must be sure to remember that the external interfaces of our open systems are not all to modern, friendly systems also designed to operate in an open environment. We must recognize that our nice open systems will also have to inter operate with existing hostile, closed systems.

Migrating from traditional very closed systems is challenging and generally requires significant effort and cost. Furthermore the cost and level of expertise required to maintain open systems is perhaps ten fold that of traditional SCADA systems.

There still is a lack of application level protocol standards. The application program interface standards supported by computer vendors differ considerably, without taking into account the more specialized real time SCADA/EMS issues. "Plug and play" in our systems will remain a challenge for a long time.

The greatest challenge for the vendors is to configure real time systems in a distributed environment within the constraints of open system standards, while continuing to meet traditional SCADA/EMS performance and availability requirements.

G. Castelli received a Bachelor of Engineering from the University of McGill in 1981. He is currently with CAE Electronics, Ltd.

Paul Emmerich received a Masters of Engineering Science from New Jersey Institiute of Technology in 1970. He is currently a Senior Prinicipal Engineer with Salt River Project.

Stanley Klein received the Doctor of Science in Operations Research from the George Washington University. He is a Principal Member Technical Staff with the Atlantic Research Corporation Professional Services Group.

Roy Hoffman received a PhD from the University of Saskatchewan in 1970. He is Manager of Control Systems Research and Development with CAE Electronics, Ltd.

Paul Traynor graduated from the University of Illinois in 1978, with a Bachelor of Science in Electrical Engineering. She is currently Manager of System Operations Engineering for Wisconsin Power and Light.

Discussion

Liam Murphy and Felix F. Wu (U.C. Berkeley, Berkeley, CA) : The desire for Open Systems Architectures for Energy Management Systems is based on the recognition that in the future EMSs will have to be 'continuously' modified, as opposed to full replacement every 15 years. However most existing EMSs consist largely of proprietary products with limited or no interoperability capabilities. Therefore, *both* in the transition to an Open Systems Architecture *and* in maintaining an EMS as an Open System, a key issue is **system migration**.

Successful system migration requires a knowledge of what products are currently available, and also of the trends in the technology and market directions. Specific EMS needs must also be considered, such as the requirements for high availability and guaranteed message arrival discussed in the paper. Equally important is the need to *decouple* the vari-

ous components of an EMS and to *standardize* the interfaces between them. This applies not only to the hardware but also to the communications system, the EMS applications, and so on. By agreeing on the responsibilities of each component and standardizing their interfaces, the effects of technology shifts (such as from CAE's IEEE 802.3 to TCP/IP) and lack of standards (such as in radio communications to pole-mounted RTUs) are localized.

These definitions would also give us one way of estimating how 'open' an architecture really is, by counting the number of available products which can be used without requiring changes to the rest of the system.

Manuscript received March 2, 1993.

Closure was not provided by the Author.