The Integrated Energy and Communication Systems Architecture

Volume IV: Technical Analysis

Appendix F:
Data Management

EPRI Project Manager

Joe Hughes

Cosponsor

Electricity Innovation Institute Consortium for Electric Infrastructure to Support a Digital Society (CEIDS)

EPRI • 3412 Hillview Avenue, Palo Alto, California 94304 • PO Box 10412, Palo Alto, California 94303 • USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

THIS DOCUMENT WAS PREPARED BY THE ORGANIZATION(S) NAMED BELOW AS AN ACCOUNT OF WORK SPONSORED OR COSPONSORED BY THE ELECTRIC POWER RESEARCH INSTITUTE, INC. (EPRI). NEITHER EPRI, ANY MEMBER OF EPRI, ANY COSPONSOR, THE ORGANIZATION(S) BELOW, NOR ANY PERSON ACTING ON BEHALF OF ANY OF THEM:

(A) MAKES ANY WARRANTY OR REPRESENTATION WHATSOEVER, EXPRESS OR IMPLIED, (I) WITH RESPECT TO THE USE OF ANY INFORMATION, APPARATUS, METHOD, PROCESS, OR SIMILAR ITEM DISCLOSED IN THIS DOCUMENT, INCLUDING MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR (II) THAT SUCH USE DOES NOT INFRINGE ON OR INTERFERE WITH PRIVATELY OWNED RIGHTS, INCLUDING ANY PARTY'S INTELLECTUAL PROPERTY, OR (III) THAT THIS DOCUMENT IS SUITABLE TO ANY PARTICULAR USER'S CIRCUMSTANCE; OR

(B) ASSUMES RESPONSIBILITY FOR ANY DAMAGES OR OTHER LIABILITY WHATSOEVER (INCLUDING ANY CONSEQUENTIAL DAMAGES, EVEN IF EPRI OR ANY EPRI REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES) RESULTING FROM YOUR SELECTION OR USE OF THIS DOCUMENT OR ANY INFORMATION, APPARATUS, METHOD, PROCESS, OR SIMILAR ITEM DISCLOSED IN THIS DOCUMENT.

ORGANIZATIONS THAT PREPARED THIS DOCUMENT

General Electric Company led by GE Global Research (Prime Contractor)

Significant Contributions made by
EnerNex Corporation
Hypertek
Lucent Technologies (Partner)
Systems Integration Specialists Company, Inc.
Utility Consulting International (Partner)

ORDERING INFORMATION

Requests for copies of this report should be directed to EPRI Orders and Conferences, 1355 Willow Way, Suite 278, Concord, CA 94520. Toll-free number: 800.313.3774, press 2, or internally x5379; voice: 925.609.9169; fax: 925.609.1310.

CITATIONS

This document describes research sponsored by EPRI and Electricity Innovation Institute.

The publication is a corporate document that should be cited in the literature in the following manner:

THE INTEGRATED ENERGY AND COMMUNICATION SYSTEMS ARCHITECTURE, EPRI, Palo Alto, CA and Electricity Innovation Institute, Palo Alto, CA: 2003 {Product ID Number

Appendix F – Network Management Technologies

Data Management Has Become the Most Complex Aspect of System Design and Implementation

1.1 Data Management Issues

The amount of data being collected or capable of being collected is increasing exponentially. This rapid expansion of data retrieval results from the fact that more field devices are being installed and that these field devices are becoming more "intelligent" both in what power system characteristics they can capture, and also in what calculations and algorithms they can execute which result in even more data.

As distribution automation extends communications to devices on feeders, as substation automation expands the information available for retrieval by substation planners, protection engineers and maintenance personnel, and as more power system asset information is stored electronically in Geographical Information Systems and AM/FM systems, even more varieties and volumes of data will be need to be maintained and managed.

Data management is a complex issue, encompassing many aspects of data accuracy, acquisition and entry, storage and access, consistency across systems, maintenance, backup and logging, and security. These are discussed in the following sections.

Data management must address a complex set of issues which include the following services:

1. Validation of source data and data exchanges
2. Ensuring data is up-to-date
3. Management of time-sensitive data flows and timely access to data by multiple different users
4. Management of data consistency and synchronization across systems
5. Management of data formats in data exchanges
6. Management of transaction integrity (backup and rollback capability)
7. Management of the naming of data items (namespace allocation and naming rules)
8. Data Accuracy
9. Data Acquisition
10. Data Entry
11. Data Storage and Access Management
12. Data Consistency across Multiple Systems

13. Database Maintenance Management

14. Data Backup and Logging

15. Application Management

No single cross-industry technology addresses all of these issues, but multiple solutions are available for different aspects. These include the best practices discussed in the following sections.

1.2 Validation of Source Data and Data Exchanges

Validation of data should be performed at each step in a process, and "flags" should be included with the data to indicate the "quality" of the data. This quality flag will normally have to be carried forward with the data to each additional step, with additional validation included whenever feasible and pertinent. For instance, power flow data from field equipment may only be validated for reasonability within a substation, but compared with other data for bias or inconsistency in a state estimation process at a control center.

Validation of data is not a technology per se, but is a theme that should run through all system designs.

1.3 Ensuring that Data is Up-to-Date

Many functions are sensitive to the time when data last updated. For these functions, a number of mechanisms are available for handling this sensitivity. These mechanisms include:

1. Time stamping data as to when it was last updated

2. Flagging data as "out-of-date" if it exceeds a specified time limit.

3. Extrapolating or interpolating data values from other more timely data

1.4 Management of Time-Sensitive Data Flows and Timely Access

Management of time-sensitive data flows entails ensuring that systems can handle the streams of data in a timely manner. Although the primary solution to this issue is to ensure that the systems and communications are appropriately sized, this sizing can entail addressing the following questions:

1. Is the flow of data relatively constant or are there bursts of data? (Queuing theory can be used to establish sizing for this issue).

2. Is maximum time of data flows critical? If so, then the system should be designed for deterministic delivery times (as opposed to statistical delivery times) and sized to handle the maximum throughput. An example is the requirement for maximum response time of 10 ms for protective relaying. In this case, non-absolute protocols like Ethernet are not feasible unless they are used in a switched network (so that essentially there never is a conflict).

3. Is it acceptable to base the timeliness of data flows on statistical characteristics? For example, access by Market Participants to market information often has statistical requirements, e.g.

95% of Market Participants must be able to access information within one second 99% of the time. The primary solution is the provision of adequate bandwidth, processor power, and well-deigned applications to ensure these contractual requirements are met. Additional solutions include alternative or backup sources of the time-sensitive information, as well as measures to prevent denial-of-service security attacks. In addition, performance logging should be implemented so that proof of meeting the statistical characteristics are available.

4. Must multiple users be accommodated with contractual commitments for access within specific time windows (e.g. access within 10 seconds after each hour, or response to a request within 30 seconds)? The primary solution is the provision of adequate bandwidth, processor power, and well-deigned applications to ensure these contractual requirements are met.

In either case, very precise specifications, statistical analysis of data flows, and rigorous factory and field testing are the most effective ways of ensuring this requirement is met.

1.5 Management of Data Consistency and Synchronization across Systems

Management of data consistency is becoming more crucial as more functions are automated, and as more decisions are made in "real-time" in which there is no leisure to check the consistency of data.

No single solution exists for ensuring data consistency; normally it is a combination of technologies and best practices applied consistently across a system. These technologies and best practices include:

1. Clear identification of primary sources of data
2. Publish/subscribe, so that systems, applications, and databases can "subscribe" to data, which will be "published" to them whenever it changes
3. Validation of data consistency through a variety of data checks
4. Alarming on inconsistencies so that they can be corrected
5. Automated procedures for establishing or re-establishing consistent data after systems are restarted or re-initialized

1.6 Management of Data Formats in Data Exchanges

Mismatches in data formats or structures are frequently a cause of incompatible data exchanges. For instance, one system will be updated so that a new attribute is added to a data item, or a new data element is deleted from a list of data. The other system usually cannot manage these mismatches in expectations, and either shuts down with an error, or blithely continues processing what it thinks it can do (sometimes correctly and sometimes incorrectly), but does not inform anyone of the inconsistency.

The use of object models to establish both the structure of each object and the contents of each data exchange can alleviate many of these problems. In addition, if metadata models of the objects are available in electronic format (e.g. as XML), then systems can automatically detect and correct mismatches.

1.7 Management of Transaction Integrity

Many systems require one-step transactions, where if the transaction fails for some reason, the results must be either ignored or rolled back. Two-step transactions are even more complex, since sometimes the results of the second transaction must also roll back the results of the first transaction.

Utility operations systems have not often needed this capability except peripherally. For example in updating the SCADA database, roll back to a previous version is needed if some error was introduced into the updated version of the database. However, this type of capability is certainly required in the market operations, and increasingly in other control center functions.

Two-step transactions and roll-back have been implemented in a number of products.

1.8 Data Accuracy

Data, whether it is monitored from the field or retrieved from some other system, needs to have appropriate precision and accuracy. In particular, distribution automation functions require more accurate data than system operators who merely need to get an estimate of conditions. This data accuracy requirement implies that (to the appropriate degree for the functions using the data):

1. The source of the data must be precise, with accurate CTs and PTs, appropriately scaled analog-to-digital conversions, appropriate resolution of the digital values (8-bits, 12 bits, or more bits to represent the analog value), and appropriately precise calculations of derived values (e.g. var, VA, kW).

2. The data sources must be accurately calibrated.

3. The data must be converted correctly to engineering units or other measurement units.

4. Data should be validated for reasonableness, if not for more precise checks. For instance, within IEDs, certain checks for consistency could be used to detect inconsistent data, while at the control center State Estimation at both the transmission and the distribution levels could be used.

1.9 Data Acquisition

Data acquisition has many pitfalls as well for data management. Some of these pitfalls can be ameliorated by new information technologies, while others require old-fashioned carefulness. The main problem is the sheer volume of data now required. Some IED controllers contain hundreds (even thousands) of points. Although many are not needed by SCADA operations, they are useful for engineers and maintenance personnel. Given this magnitude of data items, data management must be viewed as a significant aspect in the design of new and upgraded systems. In particular:

1. Object-oriented protocols (e.g. UCA (IEC61850) for field equipment, CIM for power system data, and XML for general data) should be required. These protocols require that the IED which control the field devices are organized with well-known point names which are self-describing and can be “browsed” for information much like Web pages. This self-description

allows applications to link automatically to the correct data with minimal human intervention (thus avoiding one of the main causes of error).

2. Data objects required by SNMP MIBs should be retrieved from field equipment, regardless whether object-oriented or point-oriented protocols are used.

1.10 Data Entry

One of the most difficult areas for managing data is data entry, in which humans type information into a system or database. The primary issue is trying to maintain high levels of accuracy. Again, information technology has begun to address this issue:

1. Data entry by humans should be avoided as much as possible. For instance, no data should be entered twice: a single source of each type of data should be established and used to populate other databases.

2. Object-oriented data should be used, so that applications and database tools can be used as much as possible to automate the data entry process.

3. Data should be validated as much as possible during the data entry process. This validation should include reasonability and consistency checking as well as format checking. Roll-back and two-step entry procedures should be used where critical functions must have accurate data.

4. Applications using this data should validate it as well, to avoid “crashes” and security violations.

5. Logs should be kept of all data entries.

6. Data backup of all important data should be provided.

1.11 Data Storage and Access Management

Data storage and access management is another critical area in the design of systems. Often systems have been developed for one purpose, then added-to for another purpose. Later, other applications need data, so an additional jury-rig is added. Some key recommendations for avoiding this problem (or migrating away from it) include:

1. The real-time database in the SCADA system, which is focused on providing timely but limited amounts of data to operators, should not be used as a source of data for other systems. Rather the front-end data acquisition and control (DAC) system should be structured to supply the required real-time data to SCADA system, but also provide other kinds of data to other applications without impacting the SCADA system itself.

2. The DAC should support access to field equipment by planners, protection engineers, and technicians

3. Object-oriented protocols should be used for all data exchanges between systems. With data having well-defined names, managing the access to the data is easier and more likely to be correct. These object-oriented protocols include the UCA (IEC61850), the CIM, and XML information exchange models (see next section).

4. Currently some of these object-oriented protocols are not completely interoperable or consistent in their structure. In particular, the data names and structures of IEC61850 and the CIM need to be harmonized. This activity is taking place in the IEC.

5. Standard formats and methodologies for Application Program Interfaces (APIs) for data access also need to be formalized. Currently the CIM specifies the Generic Interface Definition (GID). The GID identifies explicitly which features of existing APIs (such as DAF and DAIS) will be implemented to exchange data implemented in CIM-based databases, to extend these capabilities to include features needed in utility operations, and to specify the exact formats to use when implemented over different types of middleware (e.g. Corba or Microsoft COM).

6. Electronic registers should be developed which contain the metadata models of the object-oriented data. This is discussed in more detail in the section on Information Exchange Management.

7. As major assets are purchased, their characteristics should be entered into an electronic asset database (e.g. AM/FM system), possibly using bar codes (to avoid the fallible human data entry process). They should then be tracked throughout their life as they move from the warehouse to one or more field locations over time. This method could provide the accuracy so often missing but badly needed in the asset databases.

1.12 Data Consistency across Multiple Systems

Data consistency across multiple systems is vital for reliable automation of functions. If data is inconsistent, then the applications will have inconsistent and probably incorrect results. Many factors impact data consistency, but some methodologies can be used to help insure consistency. These include:

1. Use of publish/subscribe application services, in which every application that requires certain data “subscribes” to it. Then, whenever the source data is updated, it is “published” to all subscribers simultaneously.

2. Data should carry “quality” indications as it is passed from its source to other systems. These quality indications could include “invalid”, “out-of-date”, “manually-entered”, and “calculated”. More complex indications could include multiple timestamps indicating when the data was first created, as well as when it arrived at each database, or indications of what parameters where used in calculating its value, etc.

3. Applications should validate the input data, possibly by analysis (e.g. State Estimation), possibly by accessing multiple sources of similar data that can be cross-checked.

4. Error handling mechanisms should be in place to notify the Network Manager of loss of data, inaccessibility of data, invalid data, and other data quality indications.

1.13 Database Maintenance Management

Database maintenance is one of the most difficult jobs to perform accurately all the time. Again, the best method for managing this effort is to provide as many tools as possible for capturing the data automatically, and then verifying it before the database is released

for use by the control center systems. Many of these tools and methodologies have been mentioned in previous sections:

1. Object-oriented data using self-description techniques for correctness; eventually for automatic update capabilities

2. Cross-database consistency checking

3. Data quality indications and time-stamping

4. Updating of metadata registers and notification to applications to access this register to determine if data exchange formats have been modified

1.14 Data Backup and Logging

Data management will never achieve perfection. Therefore, critical data should always be backed up, all changes should be logged, and some means should be available to "roll back" to a previous version if necessary.

The logging requirement can also be critical for auditing as utility operations are being scrutinized in more and more detail due to the market environment.

1.15 Application Management

The management of applications is also necessary in order to ensure that functions operate correctly and accurately. Most of this management must involve the individual applications themselves, but general management methodologies can be recommended. These include:

1. The Unified Modeling Language (UML) methodology described in the previous section should be used/required for all new application development or upgrades of existing applications. Use of this methodology will help ensure that the application is properly integrated with other applications.

2. All new and updated applications should be very thoroughly tested to ensure they execute correctly under both normal and error conditions. Applications which have not been properly tested will not be trusted by users, often leading to the applications being "turned off" or ignored (this has been the fate of many power system network analysis applications).

3. The status of applications should be monitored by the SNMP Manager, with appropriate levels of notifications sent to users (e.g. alarm if a critical function is impacted, warnings if applications are "taking too long" to process data, "not responding" notifications, etc.).

This page intentionally left blank.