

Unlocking the Proprietary Mess

By James Wilk

Integral Solutions System Engineer

Who would possibly invest in equipment that restricts all their future purchases to that same initial manufacturer? Well, it happens every day and it could easily happen to you. The ability for equipment to communicate is so prevalent today most specifiers feel confident in specking it. But how does it communicate and what will understand it? Unless you are aware of the pitfalls, you could tumble in. And if you already have, this article will help you dig out.

Background

What is a power management system?

A power (aka energy) management system allows the end user to view real time data, setpoints, alarms, events, and waveforms from a centralized or distributed monitoring station. The basic components of a power management system are the meter(s), communications network and the PC (shown in Figure 1). In a simple system the PC polls the meters for data through the communications network and visualizes data in an HMI graphics package.

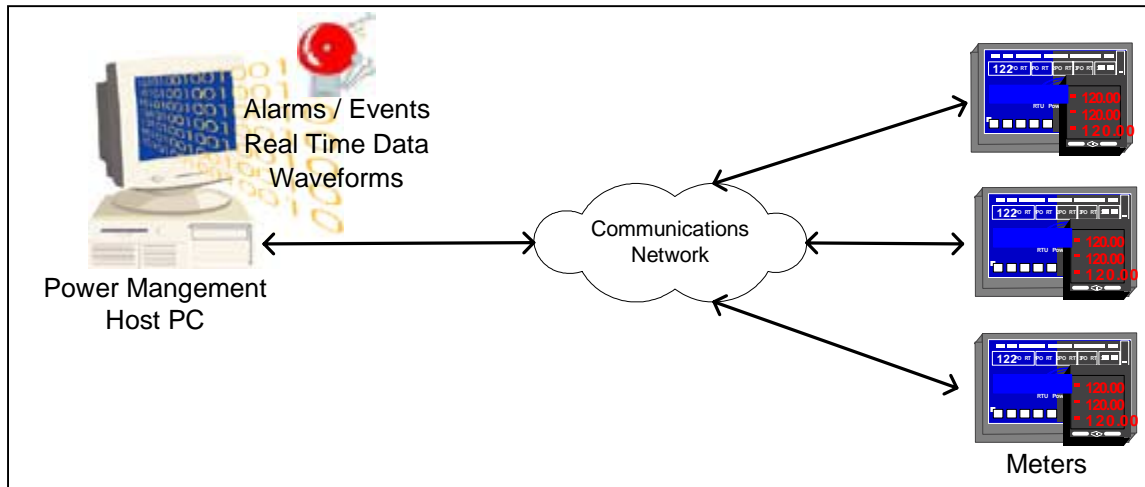


Figure 1 – Standard Architecture

The protocol is the set of rules overseeing the exchange of data over a digital communications system - the language between the PC and the meter. The PC acts as a protocol master requesting data, events, setpoints, or waveforms from the meter. The meter (slave device) responds with the information for display on the PC.

Proprietary and Open Protocol Systems

A proprietary (or closed) protocol is one that an individual or organization uses, produces, or markets under exclusive legal right. Conversely an open protocol is a non-proprietary protocol that is freely documented and formalized by an independent organization.

Modbus and DNP are prevalent open protocols seen in power management systems today because they are openly defined allowing any hardware or software manufacturers to build modbus enabled products. By adhering to the protocol rules a meter that is Modbus enabled can talk to any modbus software driver. There are three different types of Modbus – Modbus RTU, Modbus Plus and Modbus ASCII. Modbus RTU and Modbus ASCII are both open protocol languages while Modbus Plus is a special implementation of Modbus developed and controlled by Modicon. Since Modbus Plus is exclusive to Modicon it is not considered an open protocol. Modbus RTU is the most popular Modbus

type seen in automation today and is commonly called Modbus. It is good practice to verify that when a device or system says it supports Modbus, it refers to Modbus RTU. Since Modbus is so prevalent in industrial power management it will be used for the examples.

What determines if a system is truly open or proprietary? In a basic power management system three pieces of information are critical – data, alarms / events, and waveforms. If a meter communicates all of this information through an open protocol, such as Modbus, then the system is truly non – proprietary. A closed protocol system uses proprietary software or hardware to communicate with its metering and other remote data gathering or smart devices. In this way its manufacturer restricts present and future access to the system, providing themselves an economical advantage in supplying future services or equipment.

Scenario

Now that we understand the basics of a power management system and open / closed systems, lets look at a common problem illustrated in Figure 2. A company wishes to expand their existing power management system to gather data from a new building.

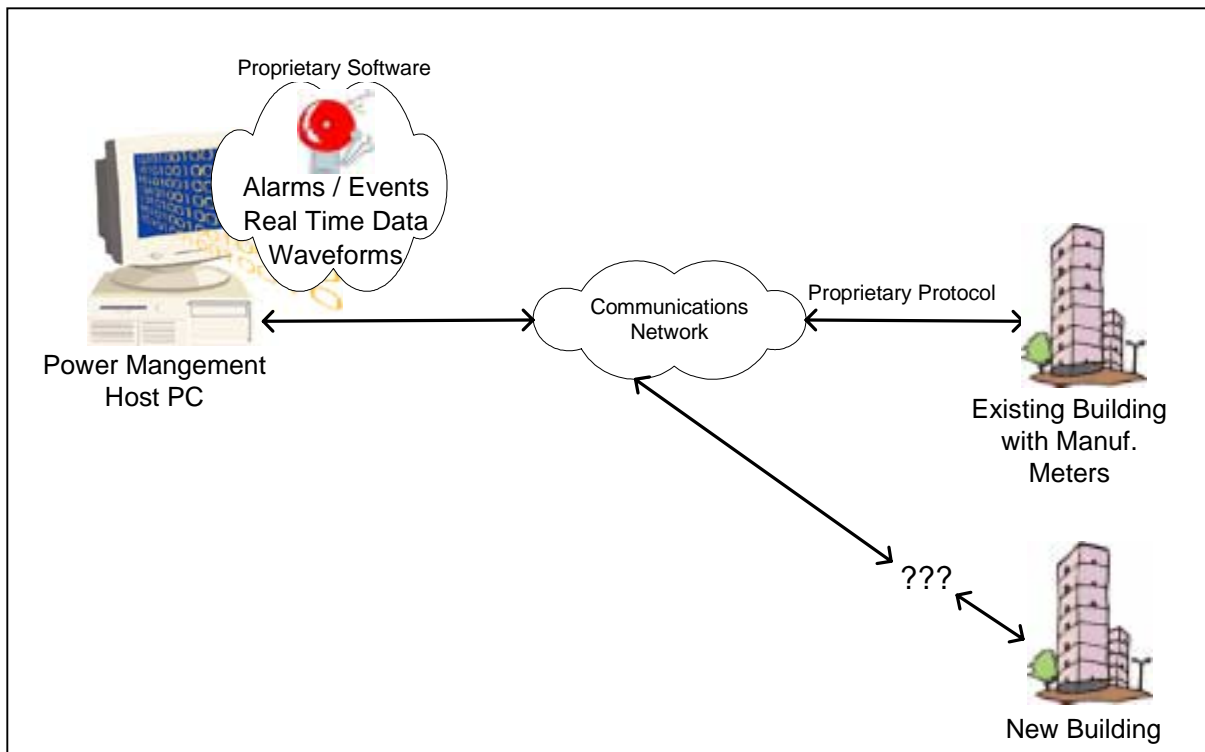


Figure 2 – What Are The Options?

While researching their options they find that only meters from the original manufacturer will work with their system. Outside meters won't work because they don't communicate the OEM's proprietary protocol. The owner is now restricted to only one supplier of equipment for his proposed new building. Joy for the manufacturer, long term ramifications for the purchaser.

The quick answer is to rip everything out and install an open protocol system. Though this approach should be evaluated, it could be very costly and could result in system down time. In order to dig our way out of this restrictive situation we need to migrate away from the proprietary protocol and open up the system at different levels. We will use modbus for the examples and list the options from the most to least drastic.

Option 1: Retrofit / Enables the meters for modbus

Some manufacturers offer a hardware or software change to upgrade a meter to talk modbus. If the meters can be upgraded, a modbus driver is installed and configured on the host PC and changes are made in the HMI.

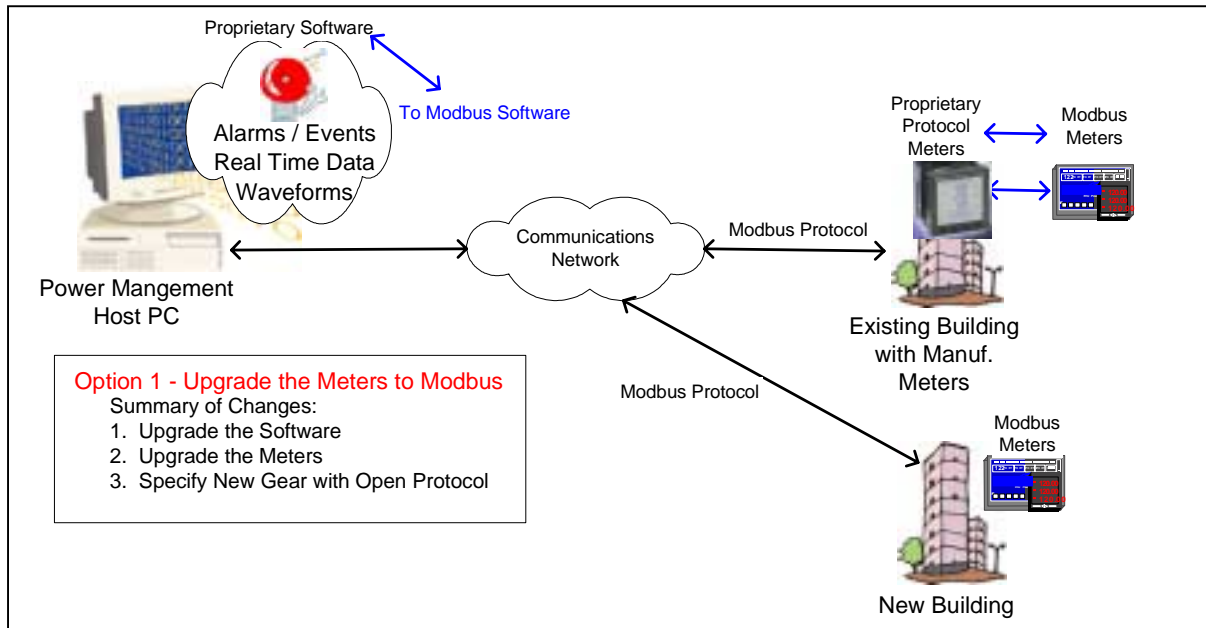


Figure 3 – Option #1 Upgrade the Meters to Modbus

Figure 3 illustrates the situation and solution. By upgrading the software, the protocol and specifying new gear with an open protocol, the customer can open up his system.

Option 2: Communication Converters

If the meter can't be upgraded, a protocol translator may be available. This piece of hardware will talk the proprietary protocol on one end and translate it to modbus on the other end. It sits between the PC and the meter and manages the communication ports and protocols.

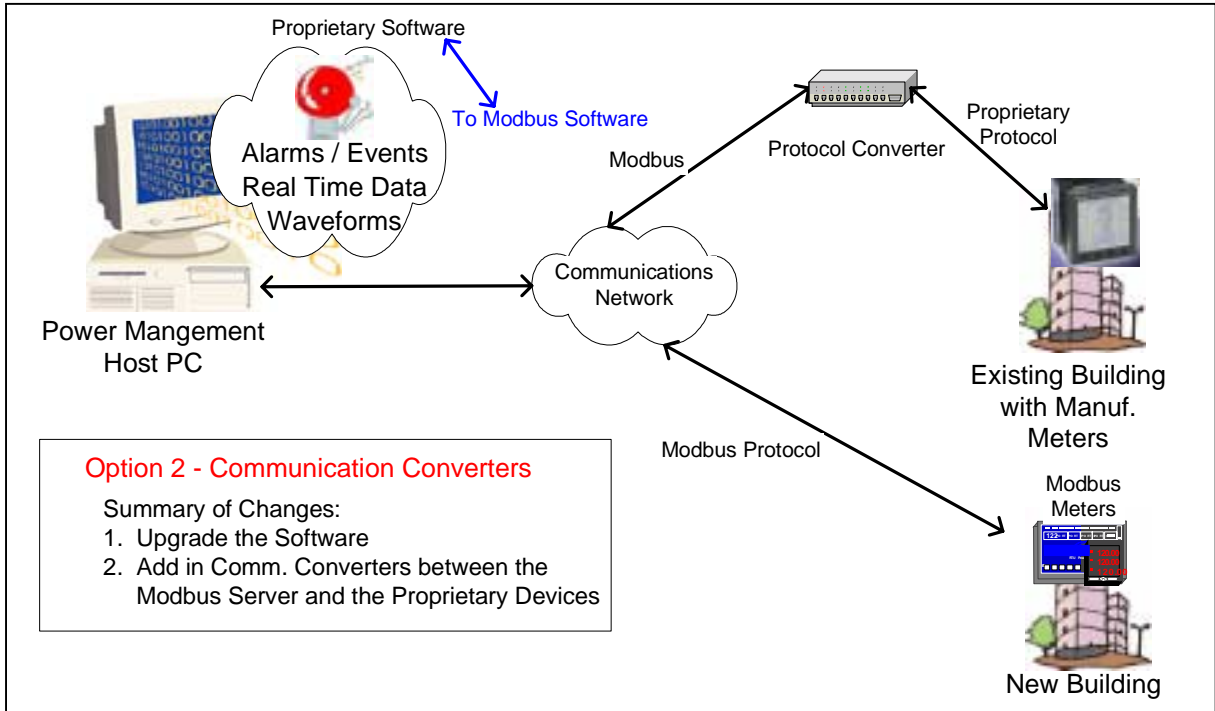


Figure 4 – Option #2 Communication Converters

Figure 4 illustrates another solution. By upgrading the software, the protocol and installing communication converters between the Modbus Server running in the PC and the proprietary protocol devices, the customer can view data from both the open and legacy devices.

Option 3: Common HMI

The least drastic solution is to leave the hardware and communications infrastructure alone and have the system run simultaneous modbus and proprietary servers with a common HMI.

By installing the new vendor's software on the same PC, the existing software will still talk their proprietary protocol and gather data from the 'old' devices. The new software will be an open protocol interface with the old software and gather data from the 'new' devices. Both will run side-by-side collecting data. In order to avoid conflicts installing and running both products on the same PC, it is important to consult and use an integrator experienced with hybrid systems.

Both servers will collect the data and provide it to a common HMI graphics display. The customer will see a unified screen and data display in the HMI, hiding all the complexity underneath. The HMI can be developed to seamlessly interface the data, waveforms, and events between the 'old' and 'new'.

Since this approach uses the recommended software for each device, the only limitations on what data can be accessed lie within the capabilities of the vendor's software. Such limitations, if any, can be addressed and resolved during the integration process.

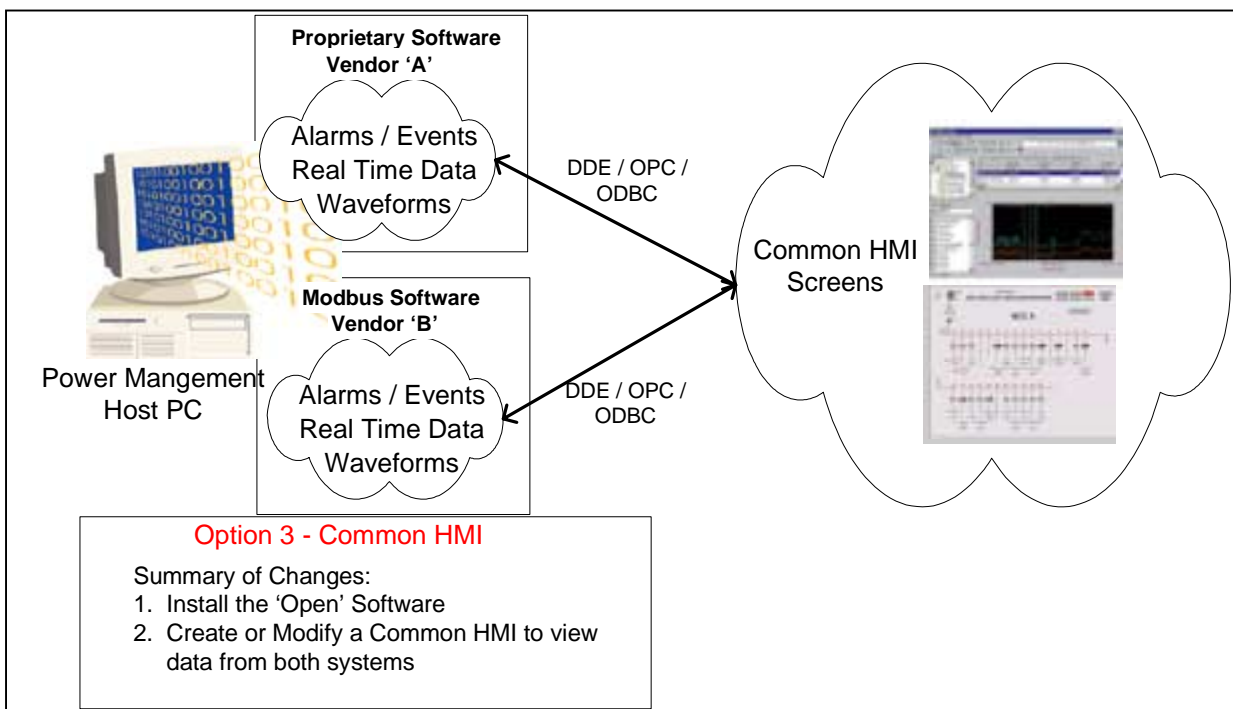


Figure 5 – Option #3 Common HMI

Figure 5 illustrates the least drastic solution. By installing the modbus software in conjunction with the legacy software each system can acquire data independently and display it on a common HMI or reporting package. In most circumstances this is the least expensive solution - with hardware costs limited to possible computer upgrades. The existing device infrastructure can remain in place with most of the work devoted to software integration.

Designing systems that won't require rescuing

Designing a system around proven open protocol hardware and software will decrease a customer's reliance on a single equipment supplier and provide flexibility to easily exchange data between any manufacturer's open protocol communicating device (meter, relay, or sensor).

The Hardware

The first step in designing a communication system that will meet a customer's immediate needs with flexibility for the future is to properly specify the hardware, architecture and software. Most manufacturers will claim their equipment to be 'open protocol.' Two very popular open protocols in substation automation today are Modbus RTU (www.modbusfoundation.org) and DNP.

The test to see if an energy management relay or meter is truly 'open' protocol is that it needs to provide information via an open protocol. Depending on the device's capabilities, it will need to provide:

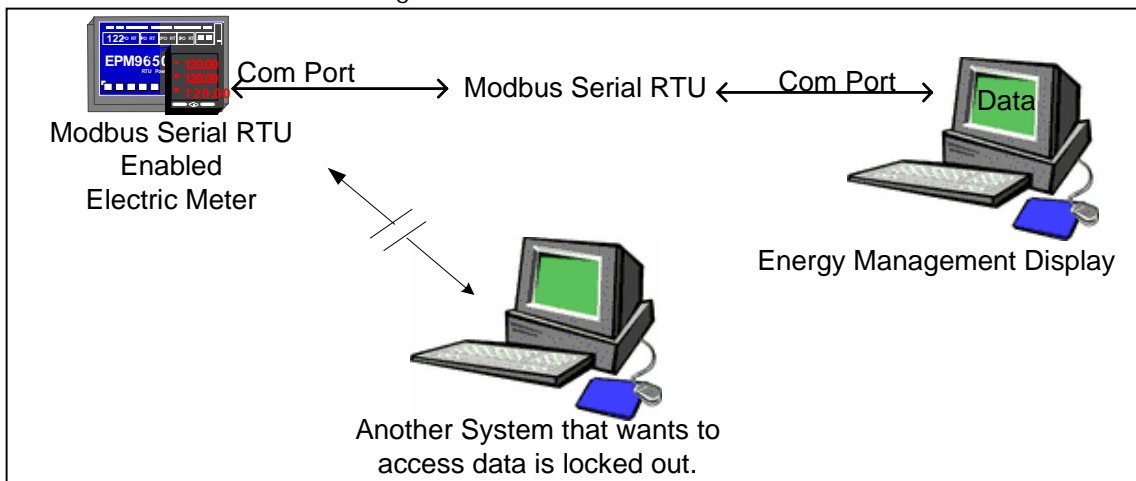
1. Metering / Setpoint Data
2. Device Events / Alarms
3. Waveforms

The procedure to acquire each piece of information from the device through an open protocol server needs to be documented and supported by the manufacturer. If a meter supports waveforms, but it requires special software to download this information, then the meter is not a true 'open protocol' device. If the manufacturer claims that events and alarms are available in a meter through an open protocol, but don't include documentation on how to access this information then the device is not 'open'.

Serial, Ethernet Copper, and Fiber are communication mediums with benefits and costs associated with each choice. A power management system using a meter's serial port will communicate Modbus Serial RTU protocol. Over a network connection it talks Modbus TCP/IP.

Ideally the system will be designed with the power management system communicating with the meter over a network connection. A meter that is ethernet ready has the ability to communicate data, through Modbus over TCP/IP, to multiple masters. For example, a serial modbus network will only allow one master to communicate with it at one time as illustrated in Figure 6 below.

Figure 6 – Modbus Serial Network



A modbus Ethernet system, as shown in Figure 7 below, would allow multiple masters to poll the same device at the same time independent of each other – opening up the system to integration with multiple third party systems without retrofitting the hardware.

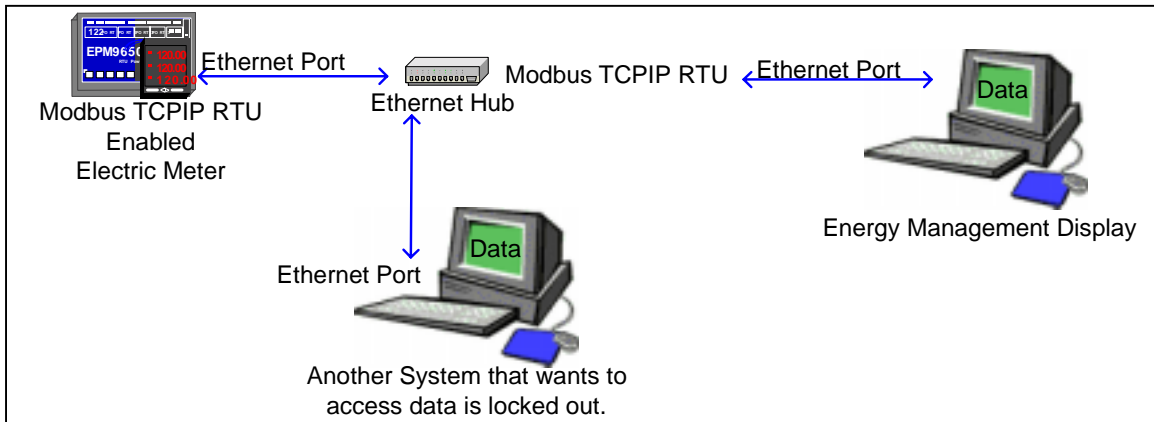


Figure 7 – Modbus Ethernet Network

The Software

A standard energy management package should include the following software modules:

Modbus Server – Used to pull the real time data and setpoints from the meter

Event Server – Automatically downloads the events in the meter

Waveform Server – Downloads the waveforms stored in the meter

It is important that the devices communicate with an open protocol. If any of the above modules saves or provides data in a proprietary file format or requires special software then it defeats the purpose of an open system and is considered closed or proprietary since it restricts access.

Access to Data

For example, client programs such as HMIs, poll the server for data to display on screens, trends and reports. The software server must support a standard protocol in order to serve this data to a client program. The traditional method of DDE communications is now being replaced by OPC. With OPC enabled software servers, multiple OPC client programs can acquire the data from the power management system running locally on the PC or on a networked node. This exposes the collected data - giving the customer the flexibility to add client nodes without having to modify the hardware or communications architecture.

Viewing Events

Logged data needs to be saved in a freely accessed format without requiring customer specific tools to effectively view and edit the information. Events and Data stored in a MS SQL database can be accessed from multiple programs for display and reporting.

Displaying Waveforms

In order to view waveforms using standard software packages the data also needs to be stored in open format. Commtrade is a popular format with many third party tools available to view and create reports using these files.

Recommendations

Any new equipment that is intended to interface with an existing power management system should include the services needed to properly interface the two. These services are referred to as integration and include the involvement of a software engineer to accomplish the requirements of the project. This involvement typically includes on-site access to, and knowledge of the existing system's capabilities and new expectations. The project specifications and drawings should address these needs, what types of equipment are to be integrated and detail the expected results. In this way the end user and specifying engineer is assured of a system, which will properly communicate, interface, and meet his long-term needs.

Conclusions

By understanding the concepts of open and closed systems, specifiers can design Power Management Systems that will meet the customer's current needs and not have him question the viability of the design in the future. For customers looking to expand or open up a proprietary system, the right approach by an experienced integrator can unlock the proprietary mess.

Appendix A – Specification Examples

The following statements can be added into a specification to help insure that an open system will be provided.

"Meters, depending on function, shall provide metering data, setpoint information, device events, and waveforms through a modbus RTU serial and / or modbus RTU TCP/IP connection".

"The Energy Management Software shall support automatic retrieval of metering data, setpoint information, device events, and waveforms through a modbus RTU serial and / or modbus RTU TCP/IP connection".

"The Energy Management Software shall automatically store device event data into a SQL database with the device's timestamp"

"The Energy Management Software shall automatically save device waveform data into comtrade format files"

"The Energy Management Software integrator shall have over 5 years of experience with Energy Management integration and shall provide, at the customer's request, proof of at least 5 successful Energy Management installations"

About the Author

Jim Wilk is a Systems Engineer with Integral Solutions - consulting, designing, and integrating substation automation systems for industrial and utility customers for the past 7 years. More information about substation automation can be found at www.integralsolutions.net