

Optimizing Supervisory Control & RFID Networks for Dynamic Information Intensive Manufacturing

Harold Clampitt and David Naylor

American RFID Solutions, LLC and Starthis Inc.
harold@americanrfidsolutions.com, naylor@starthis.com

Abstract

Adriana Karaboutis, Director, Ford Customer Order Fulfilment, defines manufacturing as “a series of material handling events and information transactions occasionally interrupted by a few seconds of value-added work.” The great insight offered by this quote is the recognition that our real task is to increase the time we spend on adding value to our manufacturing processes and assembly.

Use of technology is a key element in facilitating this goal. We explain how Radio Frequency Identification (RFID) technology elements are used to see more, do more, and get more, from existing enterprise resources. Radio frequency identification provides the low level data entry to deliver these benefits. Our paper explains the fundamentals of this technology with specific case examples for architecture within the manufacturing environment. We also discuss the challenge of integrating huge volumes of data from a pervasive and monolithic RFID layer into an architecture that accommodates both static and dynamic information. The paper explains how ambient technology gives more visibility into the process for superior control and reduction of standard deviation.

Keywords: *RFID, Radio, Automation, EPC, Middleware*

Industrial Middleware Architecture

A broad range of industry sectors use hardware, equipment and machines in the operations of their business. This equipment must be linked in some way to the systems which implement and manage the business processes of the organization. This paper focuses on techniques for general purpose industrial equipment typically controlled by programmable logic controllers (PLCs) and related items of industrial hardware. The major emphasis is on discrete and process automation, but several of the concepts apply to other sectors such as material handling in transportation, logistics and distribution. The paper also presents architectures which effectively integrate barcode and rfid data collection, since legacy compatibility is a requirement in many plants.

In order to support the high reliability and demanding MTBF requirements of manufacturing production environments, PLCs are widely used to monitor and control the sensors and actuators in industrial machines. Actuators include electrical motors, pneumatic cylinders, valves and electrical relays. Additional sensors and actuators such as buttons, rfid tag readers, bar code readers and indicator lights are used to collect or display information to control, monitor, or manage a process. Reliably managing the integration of all these elements in a manufacturing or distribution facility is a non-trivial task. A technique which has been widely adopted is to subdivide the plant into separate modules or work cells, and use a PLC as a local controller of the resources within a workcell or single machine. Within each of these cells, a PLC is tasked with managing the internal operations of the cell. This approach results in “Islands of Automation”. One successful modular approach to integrating these islands and providing a linkage to an organization’s business process management systems is to use a separate supervisory control infrastructure. The roles of the supervisory control infrastructure include:

- coordinating between the separate PLCs
- supplying the programs running on the PLCs with required input from the business systems or plant operators
- displaying and distributing information about the status of the production process
- accelerating fault isolation
- automating firmware updates to individual devices

This supervisory function is often called Supervisory Control and Data Acquisition (SCADA). In many manufacturing facilities the SCADA ‘layer’ is an intermediary between the PLCs and the manufacturing specific

business systems such as Manufacturing Execution Systems (MES), Supply Chain Management (SCM), Enterprise Resource Planning (ERP), Warehouse Management Systems (WMS), and Quality Control Systems.

The SCADA function is operated and managed by those principally responsible for plant operations. Interactions between plant operators and the process is usually performed through a graphical interface known as a Human Machine Interface (HMI) which may be provided by the SCADA application or via a separate HMI application. A common architecture for the SCADA/HMI layer which also emphasizes the daisy chain architecture is shown in Figure 1a.

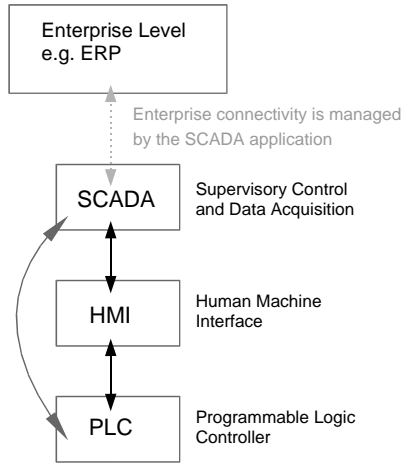


Figure 1a. Traditional SCADA architecture

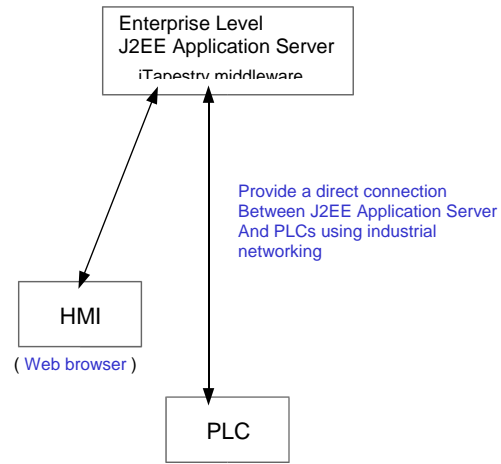


Figure 1b. Alternate SCADA architecture

Most existing SCADA applications control the flow of production related information from within the application. Examples of SCADA/HMI applications which run on the Microsoft Windows operating system include Cimplicity and iFix from General Electric, InTouch from Wonderware, RSView from Rockwell Software, and Simatic WinCC from Siemens.

Some of the challenges associated with this architectural approach include: Scalability, Reliability, Maintainability, and Security Management. These limitations result from the application architecture which is optimized for a stand alone program running on a single PC. As companies implement business processes using applications running at the enterprise or IT department level, the SCADA applications running on PCs in the plant limit the free flow of data to and from the programs running in the PLCs. Making changes to the operational and business rules implemented in the SCADA application or PLC programs can be time consuming and complex. Rapidly changing environments require a much more agile SCADA infrastructure to respond to changing needs.

A solution to solve this problem is to reengineer the SCADA infrastructure as shown for example in Figure 1b, using a distributed Service Oriented Architecture (SOA) model. There are various distributed software component environments that support the development and deployment of software comprised of separate components and services. A summary of several of these is provided in the table below. Service Oriented Architectures typically have certain common features. These include Service Providers (or Publishers), Service Brokers (or Registries) and Service Requestors (or Subscribers). Their interaction is shown in Figure 2.

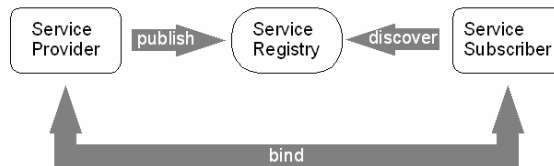


Figure 2. Key elements of a Service Oriented Architecture (SOA)

The Service Provider publishes a service description in the Registry (or Broker). A service subscriber discovers the service from the Registry and is provided some kind of “handle” with which to bind to the service provider. Examples of the Registry function and the inter-object communication mechanism for widely used distributed

software component environments are detailed in Table 1. Applications can be rapidly constructed and modified using these basic reusable service components.

Table 1. Major Service Oriented Architecture (SOA) frameworks

Object/Service Technology	Supported Platforms	Object/Service Registry	Object/Service Communications
Component Object Mode (COM)	Windows	Windows Registry	IPC
Distributed Component Object Model (DCOM)	Windows	Windows Registry	ORPC over DCE/RPC
Java RemoteMethod Invocation (RMI)	Multiplatform	RMI Registry	JRMP over TCP
Java JINI	Multiplatform	JINI Lookup Service	Application Dependent
Java J2EE	Multiplatform	Any JNDI Compatible Service	RMI-IIOP
Web Services	Multiplatform	UDDI	SOAP, RMI-IIOP
CORBA	Multiplatform	Corba Naming Service	RMI-IIOP

Besides using an SOA, there are other middleware architectures that have been developed to simplify the tasks of linking control systems, and data collection devices with management applications. These middleware architectures allow devices and applications from different vendors to be integrated, just as with the SOAs, but they do not follow the structure shown earlier in Figure 2.

Table 2. Additional industrial automation middleware architectures.

Architecture	Organization	Brief Description
OPC	OPCFoundation [1]	Eliminates custom driver interfaces between SCADA applications and automation devices by providing a common client interface onto a process and device data repository.
Interface for Distributed Automation (IDA)	IDA Group [2]	Includes the RTPS (real-time publish-subscribe) protocol, which is based upon the Network Data Delivery Service (NDDS) from Real Time innovations, Inc. The IDA organization has more recently focused specifically on the Modbus protocol.
MQ Series Integrator SCADA Device Protocol (MQIsdp)	Arcom Control Systems And IBM [3]	A lightweight TCP/IP based protocol to connect remote devices and data producers with IBM's WebSphere MQ Integrator. The protocol uses a publish and subscribe model and links into the MQ Series enterprise messaging middleware, which is a similar messaging oriented middleware.
Open System Architecture for Enterprise Application Integration (OSA-EAI)	Machinery Information Management Open Systems Alliance (MIMOSA) [4]	The MIMOSA standards are intended to provide a dynamic, flexible framework for building Operations and Maintenance applications that are compliant with the ISO 13374-1 standard for machinery diagnostics systems.

The most widely used of these automation middleware architectures, OPC helps solve the problem of device driver development, but as a middleware architecture it provides a limited set of capabilities. When used in conjunction with one or more PLCs, it maintains a copy of the state (values) of tags in the PLC program. A benefit of this approach is that when large numbers of OPC clients require access to data, the values can be retrieved from OPC server data cache without burdening the PLC processor with multiple requests for the same tag values. This has the possibility of reducing the communication burden on the PLC. Conversely if the OPC server is supplying a small number of OPC clients with infrequent data requests, the OPC server places an unnecessarily high communication burden on the PLC while reading data values that are not needed.

Another limitation of the OPC server is that it is a Microsoft Windows specific technology. The OPC-DA XML standard does allow for a cross platform OPC client or server, but at the expense of substantial reduction in performance due to the XML parsing requirements of the messaging scheme. One of the benefits of OPC is its flexibility and ease of use, but this is achieved at the expense of a limited set of capabilities. Specifically while it provides a mechanism for storing device object property values, it does not provide a way of calling device object methods, nor does it support sophisticated messaging schemes such as the CIP (Control and Information Protocol) Routing used in the Ethernet/IP industrial automation protocol.

The architectural approach taken at Starthis with the iTapestry middleware has been to migrate the SCADA layer to the same platform as used by many manufacturing enterprise application, specifically the Java 2 Enterprise

Edition (J2EE) platform. The benefits of running the SCADA middleware on J2EE rather than on an intermediate platform such as a Windows PC running an OPC server include:

- Operation on an enterprise strength reliable platform
- Support for server clustering (for Continuous Availability).
- Better integration with robust enterprise databases
- Integration with centralized Identity Management Services
- Platform independent
- Centrally managed
- More scalable & more secure

Not only does the application server provide an efficient platform for integrating with centralized IT resources, but it also can serve traditional plant floor operator screens from a centralized standards-based secure platform, as shown in Figure 3.

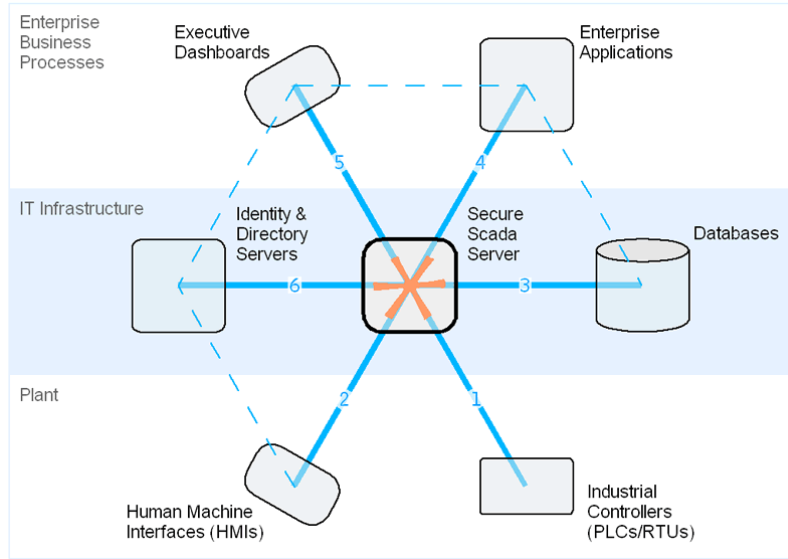


Figure 3. Consolidating SCADA functionality onto a secure robust server integrated with enterprise services

Linking into the Authentication and Authorization services available on the J2EE platform allow actions triggered by an operator to be individually authenticated and authorized. This is a requirement for food and pharmaceutical companies covered by FDA regulation 21 CFR Part 11. Many of the benefits of this architecture do not come from the middleware software itself as much as they come from the other robust services on the J2EE platform which can be easily accessed and leveraged in this SOA approach.

The middleware itself is composed of two kinds of software components:

- J2EE Connector Architecture (JCA) connectors which provide support for the industrial automation protocol (e.g. Ethernet/IP or Modbus/TCP) and
- Enterprise Java Beans (EJBs) which provide a service-oriented software interface onto the capabilities of the automation devices such as the PLC.

The EJBs provide a convenient software interface for efficiently transferring data to and from the tags in the PLC via the native data formats used within the industrial automation protocol. If the PLC is controlling a machine, or a process, additional customized EJBs can be produced easily which expose the functionality of the machine rather than providing just a low level tag oriented interface.

Integrating different types of automation systems such as PLCs and networks of Ethernet/serial devices, like barcode readers and barcode printers, in a way that is flexible and easily modified, can be challenging. One approach is to use the PLCs as the data management hub. Another is to use the PLCs for local logical control, and then have a separate networked infrastructure for the information oriented devices. The integration between an MES system at the IT level, and the plant operations is then achieved, not by a middleware layer linking the MES with tag data in the PLC, but rather by using the barcode or rfid read events as a mechanism for knowing when individual events have occurred in the plant. This provides a loose, but flexible linkage between the MES and the plant operations.

While PLCs have developed some general purpose information processing capabilities, their greatest strength is in highly reliable localized automatic control. Information processing is most usually performed at the SCADA level. As a result, rather than connecting the readers to the PLC as shown on the left, an alternate approach which leverages SOA is to use a parallel middleware for the barcode and rfid readers. This is particularly important when the middleware must integrate with networked information services such as those used in the EPCglobal network [5]. It should be noted that the EPCglobal network services themselves follow the model shown earlier in Figure 2. The registry used in this case is the EPC Discovery Service.

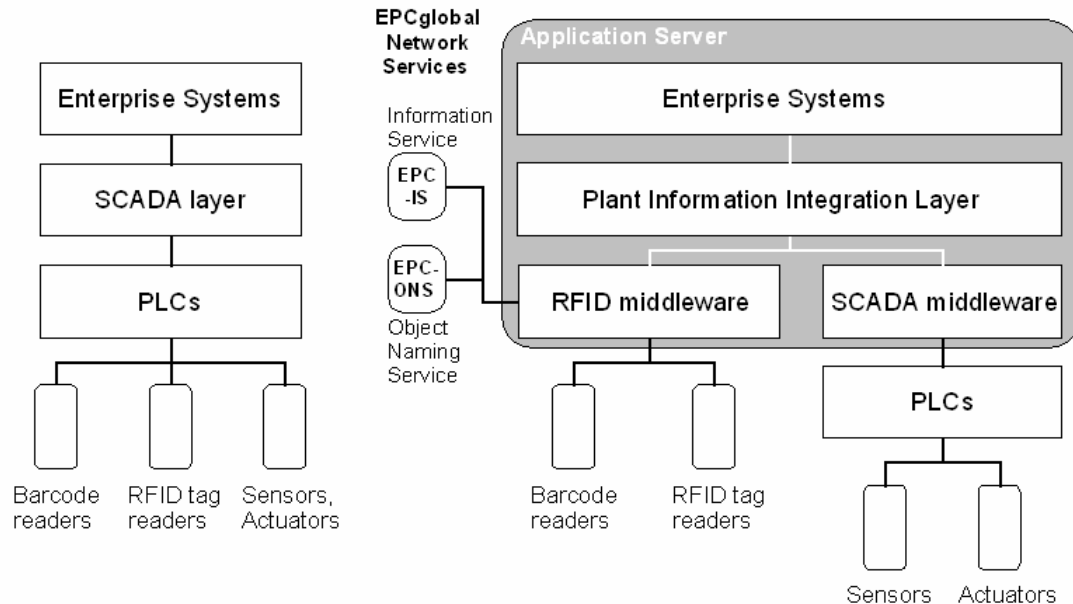


Figure 4. A PLC-centric data collection environment vs a service oriented environment

While the architecture on the right may at first seem more complex, it is possible for the upper three layers in the figure to all be deployed on a single application server. RFID middleware products from vendors such as OAT Systems, ConnecTerra, IBM and Sun Microsystems can all be deployed on J2EE Application Servers. As business requirements change, they can be implemented in the centralized Plant Information Integration Layer, rather than requiring reprogramming of many individual PLCs.

RFID Benefits

One of the most leverageable benefits of RFID is the ability to deploy a monolithic architecture of “nanoblock” integrated circuits with 2k+ bits of memory [6]. RFID’s physical layer provides a wireless infrastructure to spatially and temporally increase the resolution by which we measure, monitor and analyze the world we live in. In other words, RFID is delivering ambient intelligence to increase visibility of events, processes, triggers, inputs and outputs. It does this by placing RFID tags onto goods, automation machines, locations and people. Through the use of tags, readers, antennas and middleware, the RFID infrastructure delivers empirical data to build powerful dynamic models of the environment. Traditional static databases will become extinct as they are gradually replaced by a hybrid database of variable and dynamic data.

Within the manufacturing world, RFID systems yield a huge volume of data to describe a manufacturer’s work in process inventory at the item level. This was not feasible prior to RFID, the Internet and Wireless. Just how much data are we talking about? Consider Wal-mart’s use case to gain a sense of the enormity of the information. Wal-mart requires its suppliers to adopt RFID and this will ultimately require manufacturers to tag products at the source of production. According to Table 3 below, Manufacturers need to supply Wal-mart with 60 Billion items a year for item level tagging of its products. As RFID market share increases towards the 2 trillion bar codes produced each year you gain a huge installed base of wireless integrated circuits which are capable of logic and memory [7]. Ultimately the deployment of RFID technology will be ubiquitous and therefore provides a large enough installed base to have ambient intelligence. Or in the alternative, RFID is on stuff everywhere.

Table 3. Item level Tags in Wal-Mart Stores [8]

	Discount Store	Supercenter	SAM's Club	Neighbourhood Mkt
SKUs	80,000	100,000	4,000	28,000
Items per SKU	25	50	100	25
RFID Tags per store	2,000,000	5,000,000	400,000	700,000
# of Stores	1353	1713	526	85
Tags in Chain	2,706 Million	8565 Million	210 Million	60 Million
Annual Tags	16 Billion	51 Billion	1.3 Billion	360 Million

Tracking and analyzing the RFID data is a whole new area of software application that also did not exist before -- certainly not on any sizeable scale. RFID analytics is different from Middleware and far more agile than the traditional Business Intelligence offered from standard ERP systems or specialized software from companies like Cognos (www.cognos.com). This is not just about getting tag data out of a reader and into the correct fields in an ERP system. Nor is it seeing data trends in a multi-regression analysis with auto correlation data mining being cleverly applied. Real time RFID analytics will recognize and use the distributed intelligence to make local decisions at the cell or machine level in a factory which allows for autonomous command and control in real time.

As a result of these benefits we will have a new field of RFID Analytics that performs computations on hybrid databases which will be governed by real time enterprise rules to optimize business decisions. What do we mean by a Hybrid database? It is a new paradigm where the source of information in a database is demarcated by dynamic real time information provided by RFID and the static information provided by the enterprise to define the product. The real time information for a product may be its birth date, plant of origin, cells or machines used in each of the value add steps needed to create the product whereas the static information may be the SKU, bill of material cost, specifications and other product information used by the enterprise to define the product or understand its DNA. Combining both types of information will allow an enterprise to optimize its wealth creation by applying business rules with context in real time. This is the most significant derivative from RFID, especially as the number of readers that are networked increases.

Not only do manufacturers need to consider upgrading their plant data networks to handle larger amounts of data from individual item level tracking, but the knowledge obtained from this data will allow them to operate more nimbly if they have the architecture in place to respond rapidly to an ever changing business environment.

Summary

Industrial middleware offers optimum benefits when real time control vis a vis business processes are coordinating production to satisfy demand. In this paradigm we create high stakes dynamic information flow by architecting a direct connection between the J2EE Application Server and the following device beans: HMI, PLC, Barcode Readers and RFID Readers. As the number of RFID tags proliferate so does the number of readers and other devices. The net effect is to increase the quantity of network nodes possessing logic and memory. This is akin to adding more intelligence into your manufacturing system. If this intelligence is applied to creating new benefits like self healing industrial machines, dynamic control of cells and automated exception reporting then RFID has become "the rising tide that lifts all boats". With this architecture it is possible to increase the resolution for product/process visibility while simultaneously allowing RFID analytics to learn more, do more and see more. As a result, the material handling events have been streamlined and the information transactions are superior in delivering more time for value added work.

References

- [1] <http://www.opcfoundation.org>.
- [2] <http://www2.modbus-ida.org/idagroup>
- [3] <http://www.arcomcontrols.com>
- [4] <http://www.mimosa.org>
- [5] 2004, "The EPCglobal Network: Overview of Design, Benefits, & Security", EPCGlobal Inc.
- [6] Alien Technology nomenclature for its RFID integrated circuit with 2kbit of memory in a 350 micron square integrated circuit. Texas Instruments, Philips, Symbol and Impinj have equivalent RFID infrastructure.
- [7] Technology Review (Cambridge, Mass.), April 2003 v106 i3 p80(1)
- [8] ValueLine Reports, Wal-mart 2005 Annual Report, Form 10-K & 10-Q, Edgar Online