WinRFID – Middleware for Distributed RFID Infrastructure

B.S. Prabhu, Xiaoyong Su, Charlie Qiu, Harish Ramamurthy, Peter Chu, Rajit Gadh
Wireless Internet for the Mobile Enterprise Consortium, UCLA

Introduction

Computing technologies have been evolving rapidly, but sophisticated, speciality systems – hardware, system software or application software - remain distinct, heterogeneous, and cater to the solving of critical problems.   Middleware is the software paradigm employed by industry to assimilate these technologies into the established computing infrastructure with minimal disruptions[1].

Radio Frequency Identification (RFID) technology is one such technology which is perceived to benefit many industry verticals.   RFID would impact by way of improving asset visibility and management, better security and safety, process simplification and facilitate alternate process execution models[2].   However, integrating and using RFID in many of the existing operations has been found to be rather challenging on account of diverse hardware, evolving standards, amount of data and information generated, and lack of support for RFID operations and data in existing IT infrastructure – a prime candidate for middleware based solutions[3].

RFID research at WINMEC, UCLA

The focus of the RFID research program at Wireless Internet for the Mobile Enterprise Consortium (WINMEC) at UCLA has been to promote, educate, guide and show-case the potential of RFID technology across multiple industry verticals[4].   WinRFID, an ongoing middleware technology being developed in-house demonstrates the ease of integration of the technology into existing IT infrastructure and using the data for better decision making.   The challenges that WinRFID attempts to mitigate are technology integration, hardware management, control high-volume real-time data exchange, convert unintelligent data into decision knowledge and format the decision knowledge for consumption by enterprise applications.
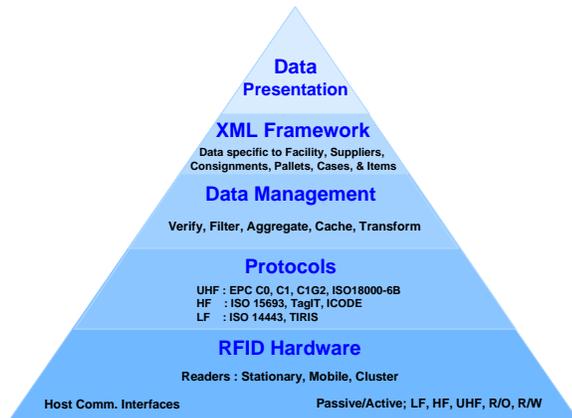
WinRFID – RFID Middleware

WinRFID - edge-of-the-network – middleware is being developed on Microsoft .NET framework[5]. It provides a distributed environment to process the data from tags read by the readers, translates the data where necessary, and routes it to a variety of backend applications using suitable technologies such as Web, Remote and Windows Services.   The middleware has a set of application programming and integration interfaces for supporting application development. At same time, WinRFID has plug-in capability to accommodate symbiotic technologies such as location services and sensors, which would not only enhance the technological effectiveness of the RFID system, but also add value to the business processes[6].

To deal with high volume data, WinRFID middleware is supported by novel algorithms and data representation schemes capable of processing large amounts of data, rectifying errors in real-time, identifying patterns, correlating events, reorganizing and scrubbing data and recovering from faults and exceptions.   Interoperability involves simultaneous distributed working of receivers/readers and transponders/tags at different frequencies using different protocols, with read/write capabilities, different read rates, and other characteristics as a layer transparent to the applications.   Network management involves deployment, initialization and control of receivers and transponders, which can be organized into a hierarchical structure with

operational syntax and semantics attached to each or a group of receivers, transponders and concentrators or even the edge computers.

Architecture of WinRFID

WinRFID has five main layers.   The first layer deals with the hardware – readers, tags and other sensors.   The second layer abstracts the reader-tag protocols. Above that lies the data processing layer, which deals with processing the data streams generated by the reader network. Fourth layer constitutes the XML framework for data and information representation. The top layer deals with the data presentation as per the requirements of the end-users or different enterprise applications.   The figure depicts the different layers.

Physical Layer – Hardware

This layer deals with the abstraction of three elements of the RFID infrastructure - readers, tags and host I/O interfaces.   The abstraction makes it very simple to derive any new specific reader, tag or I/O interface to extend the middleware capabilities in the advent of introduction of new RFID technology.

The reader object assists in management, configuration, location assignment, associate tag protocol(s), security, and the interface for command engine or the dedicated API/SDK provided by the vendor.

Data
Presentation

XML Framework
Data specific to Facility, Suppliers,
Consignments, Pallets, Cases, & Items

Data Management

Verify, Filter, Aggregate, Cache, Transform

Protocols
UHF : EPC C0, C1, C1G2, ISO18000-6B
HF   : ISO 15693, TagIT, ICODE
LF   : ISO 14443, TIRIS

RFID Hardware

Readers : Stationary, Mobile, Cluster

Host Comm. Interfaces   **Passive/Active; LF, HF, UHF, R/O, R/W**

WinRFID - multi-layered architecture

Protocol Layer

In a comprehensive RFID middleware, support for multiple tag protocols and the capability to add new ones as they become available is imperative. In WinRFID, the protocol component is also abstracted to wrap the command syntax and semantics of a variety of published protocols such as ISO 15693, ISO 14443, ISO 18000 – 6 A/B, ICode, EPC Class 0 and EPC Class 1.   It deals with protocol specifics such as byte-based, block or even page reading and writing, structure and length of the command frames, partitioning of the tag memory space, checksums, etc.

Data Management

Problems due to tag density, read/write distance, orientation of tags and material of item introduce inconsistencies in reading or writing such as multiple reads of the same tag, some tags not being read, erroneous reads, etc.   These issues are addressed in this layer by having processing rules to weed out duplicate reads, verify the tag reads, and when advanced records are available such as advanced shipping notices, this layer reconciles the records with the tag reads.   Any discrepancy is processed as exceptions and a variety of alerting systems are available for resolution – emails, messages, or user defined triggers.

XML Framework

The raw cleaned (verified and filtered) tag data from the physical layer data streams is formatted in a variety of ways to a high-level XML based representation. The information is filtered, cleaned, aggregated and adapted as per the custom plug-ins, which can be added to

the middleware services. The attempt is to provide data in a format amenable to decision making at the application layer.

Data Presentation

This layer facilitates data visualization for decision making. Currently, the portal and the database connectors are supported. The portal allows users to subscribe to the information of interest. The data delivery format can be default as provided by the middleware or the subscriber can register data adapter plug-ins. All such plug-ins would be available through a library. Other features of the portal are plugging the RFID data into graphic visual widgets (charts, graphs, etc.) for presentation. From each of these widgets the portal will let the subscribers to make decisions like trigger events for re-routing, re-assign, billing, alert, etc.

The other connector is the database connector. Currently the middleware can populate SQL Server and Oracle RDBMS. The databases get populated in an asynchronous fashion in a trickle mode – a process with least priority so as to avoid the edge hosts getting locked up. Priority of the resources is skewed towards processing the activities of the lower three layers as shown in figure 5 and the upper layers being catered to in the background at lower priority.

Rule Engine

In WinRFID, large sets of data from multiple sources (readers and sensors) and changing status (standards, protocols) of the RFID technology is handled by a tightly coupled rule engine into the architecture of WinRFID. An attempt is made to craft the architecture in a flexible way to the extent that it provides a means to even the end users to incorporate their own rule chunks by way of 'plug-ins'. The rule engine influences a number of processes and activities of WinRFID, such as raw data filtering, aggregating, exception handling and alerting at the edge node, and data adaptation, options to publish or subscribe the data to and from the enterprise applications and others are all driven by the rules.

Summary

WinRFID is an RFID-technology agnostic middleware and a holistic distributed application. The design of the architecture is federated with the functional, system, business and process knowledge residing in self contained software units - the different services providing a variety of independent and complementary capabilities.

WinRFID infrastructure technology is being tested for a diverse set of applications including medical/healthcare, asset tracking, supply chain, retail, library/document tracking, security and e-pedigree[7,8].

References

1. Gadh, R., Oct. 2004, RFID: Getting From Mandates to a Wireless Internet of Artifacts. *Computerworld*, Oct. 4.
2. Gadh, R., Aug. 2004, The State of RFID: Heading Toward a Wireless Internet of Artifacts. *Computerworld*, Aug. 11.
3. IBM Wireless e-Business Group, 2003, RFID in Business Processes. A Thought Leadership Technical Paper.
4. RFID@WINMEC [http://www.wireless.ucla.edu/rfid/].
5. B. S. Prabhu, Xiaoyong Su, H. Ramamurthy, Chi-Cheng Chu, Rajit Gadh, WinRFID – A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications " Invited chapter in Mobile , Wireless and Sensor Networks : Technology, Applications and Future Directions, Rajeev Shorey, et al., John Wiley (under publication).
6. Ramamurthy, H., Prabhu, B.S., Gadh, R., 2004, Reconfigurable Wireless Interface for Networking Sensors (ReWINS). Personal Wireless Communications: IFIP TC6 9th International Conference, PWC 2004 [LNCS 3260 / 2004], eds., Niemegeers, I., and Heemstra de Groot, S., Delft, The Netherlands, Sep. 21-23.
7. SpecimenTrak [http://www.winmec.ucla.edu/inventory/]
8. RFID Applications [http://www.wireless.ucla.edu/rfid/winrfid/demos.asp]