Generic RFID Application Framework

by

Mohammad Zuber Khan

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

June 2008

Generic RFID Application Framework

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

Mohammad Zuber Khan

to the

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY,KANPUR

June 2008

Certificate

This is to certify that the work contained in the thesis titled "Generic RFID Application Framework", by *Mohammad Zuber Khan*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

June, 2008

(Dr. Rajat Moona)

Department of Computer Science & Engineering,

Indian Institute of Technology Kanpur

(Dr. A R Harish)

Department of Electrical Engineering,

Indian Institute of Technology Kanpur

Abstract

Radio frequency based Identification (RFID) is one of the most exciting and promising technology in the field of automated identification. It is expected to be deployed in a major way in the near future. The major issues for such a deployment are in the design of a robust and flexible software system to interface various applications to the RFID readers. There are a few existing RFID software systems available that fulfill these requirements to a certain extent, however, most of these are proprietary. These proprietary RFID software solutions are costly, bulky and non-portable to different environments.

In this work, we present an open-source ‘RFID Application framework ’ which is simple and easy to use. This framework allows us to create RFID applications without worrying about the hardware and protocol intricacies. It allows us to interface multiple applications with the RFID tags and readers in a technology-neutral (protocols, air-interface, etc.) manner. The system supports multiple applications simultaneously interacting with one or more readers or even a part of the reader. It is possible, to build applications having different analysis and processing requirements using this application framework.

In this work, we have developed and analyzed two applications for electronic file tracking and tracking of postal bags .

Acknowledgments

I wish to thank my thesis advisers, Dr. Rajat Moona and Dr. A R Harish for all their support and guidance. Their encouragement and enthusiasm has been my primary source of inspiration and motivation. It was indeed an honour to work under them. Without their encouragement, this work would have not been possible. The weekly discussions and reasoning for different approaches made an enjoyable experience for this work.

I thank Prof. Veena Bansal for her invaluable advice in my thesis. Her inputs regarding the resistance to change in an organization has been very helpful to understand and analyze the problems involved in the real world deployment of an RFID system, helping me to develop a user friendly system.

I thank Anirudh Ghayal, my classmate, for being actively involved in all phases of my thesis. The various suggestions during these meeting helped at various stages in this work. I am also thankful to all my MTech friends for constantly encouraging and supporting me throughout my stay here at IIT Kanpur.

I thank the entire Computer Science Department for making such an enjoyable place to work. The support of the administrative and technical staff helped in easy access of resources as and when needed.

Finally, I am grateful to my parents and all my relatives for all their love and support. Their constant affection and encouragement has helped me achieve my goals.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

RFID systems have been used quite extensively for various types of applications which involves tracking [22], identification [40], access management [38], etc. These applications have to interface with different RFID hardware devices – readers and tags, to get data. RFID devices have different connection-interfaces to connect to the system. Different tags and readers use different protocols to communicate often in different frequency ranges. The porting of an application becomes very difficult when it is migrated from one hardware to another. This poses a great hindrance to wide adoption of RFID in small and medium enterprises as the cost of porting becomes very high.

There are a large number of parameters[30] of the RFID hardware which have to be configured before they can be used efficiently. These parameters include various kinds of antennae to be used, the power level, the communication parameters, etc. To configure these parameters, the application has to be aware of the RFID device it is using.

An application also has to apply different logical rules to infuse intelligence into the system. It has to actually process the data and present the data to the end-consumer in an understandable manner.

All these requirements from an application put a huge cost to develop RFID based applications in terms of knowledge and time. There is a need to separate out the hardware related processing and configuration from the application level processing and presentation. This will lead to effort being channelized to the development of applications with sound business logic and efficient processing and presentation. It will also lead to easier and faster development of RFID based applications.

1.1 Problem Statement

In this thesis, our goal has been to design and implement a generic software framework to be used for the development of various RFID applications.

The goals we seek to achieve in this work include the following.

- Design of an application framework, which is generic enough to support various application requirements.
- The framework should provide an easy mechanism to configure various middleware services.
- Development of complex real world applications, using the framework to define various hardware requirements and to define the logical rules.

We have designed and implemented a generic RFID application framework which provides a graphical user interface (GUI) to configure and use various RFID readers through an RFID middleware. This framework also allows the applications to define various events and their rules, in order to make sense out of the data read by the readers.

We have built two real world applications – Electronic File Tracking System and a Postal Article Tracking System. We use the application framework to handle various hardware configuration requirements and data handling. The applications are provided with the requisite data, as configured by the applications and the application applies the predefined logic rules to figure out occurrence of various events. These events are then presented to the end-consumer in an understandable fashion.

1.2 Related Work

RFID provides an efficient way of automatically identify the objects. This property of the RFID devices has enabled it to be used in many applications concerning identification and tracking. One of the most widely used application is the access control systems, where RFID based plastic cards are used to identify and authenticate the card-holder's entry to the facility. The RFID systems are extensively used in the warehouses and stores for the supply chain management [18, 32], inventory management [21] and movement management [22, 24]. This has led to huge increase in the efficiency of the warehouse operations and keeping the optimum inventory in the stores. Here we discuss some RFID based applications.

1.2.1 Library Management

Various operations in the library management [20] includes circulation, shelf management and sorting of books. In the current mode of operation, the barcodes are attached with the books which are used then to identify a book. The circulation and sorting still requires a lot of human intervention. However, the bar code does not help in anyway in shelf management and theft detection.

RFID presents a very good solution for library management. The various advantages this system leverages to the library, can be seen in circulation, shelf management and sorting of the books. Another feature that is made available by using the RFID technology is theft detection. In case of a library, books are circulated several times, making the RFID tags to be used several times. This reduces the cost of operation as well.

The use of RFID reduces the amount of time required to perform circulation operations. In the case of bar codes, a lot of time is consumed in reading the tag as each individual book has to be brought sufficiently closer to the bar code reader. However, by using the RFID, the information can be read faster as information from several books can be read even if they are kept in a stack, making the circulation handling faster.

RFID tags provide great advantages over other identification systems, like barcodes, because the tags can be read while the books are placed on the shelf. This feature significantly reduces the cost and time involved in handling inventory of the library. With regular inventory of the shelves, the books may be tracked if they are placed on a wrong shelf.

The presence of RFID tags on the books also provides for theft management to the library authorities. This can be achieved by placing antennae at the exit points of the library, and raising an alarm when an unissued book is taken out.

1.2.2 Supply Chain Management

In today's retail market, accurate identification and tracking of goods is essential [35, 18]. More than ever before, there is pressure on manufacturers, distributors and retailers to maximize efficiency, minimize cost and provide the best possible value to the end-customer.

The RFID technology can be utilized to make large quantities of information analyzed and made available to internal and external systems in near-real time, which is crucial for improving the quality of business operations.

The information on tags is read when they pass by an RFID reader. The movement is therefore captured and managed by the business infrastructure. The infrastructure includes database servers and other business rules, etc. In this way, organizations are able to link the physical world to the digital world without any human interaction. Whatever actions are then triggered depends on the individual application, from basic stock replenishment at one end of the spectrum to facilitating the ultimate lean supply chain at the other.

Thus, RFID enables the enterprises to achieve the following.

- Improved tracking and visibility of high-value items from the source to the destination.
- Reduced errors in shipping of goods to wrong places.
- Inventory visibility and efficient stocking of goods.
- Improved production planning and smarter recalls.
- Reducing counterfeit products.

1.2.3 Health Care

In the health care space, there is application of RFID in pharmaceutical industry [33] as well in the hospital industry [36]. The adoption of RFID technology in these health care industries has been of immense help to the end-consumer and has lead to increased profit for these industries [8].

The important applications in the pharmaceutical industry include the following.

Drug Counterfeiting

RFID tags help in detecting products that are counterfeit or fake. It helps to identify drugs that are tampered, adulterated or substituted.

Clinical Trials

RFID technology improves the tracking of drug usage throughout the clinical-phase testing protocols. Improved tracking and accountability can improve the reliability and speed drug approval process.

Inventory Management

Increased inventory visibility could reduce buffer stocks by substituting knowledge for inventory, thereby reducing total inventory costs.

Some of the applications for the hospital industry are the following.

Medical Device and Asset Tracking

Surgical instruments and other devices must be properly cleaned and packaged between uses. Tags on the instruments and readers on the sterilization chambers and storage cabinets can validate proper cleaning and help locate needed instruments. Since medical devices are often mounted on portable carts, smart tags placed on the devices and readers installed in the doorways can enable personnel to quickly locate a crucial piece of equipment and immediately determine its fitness for use.

Patient Tracking

Patient identification and location assistance are often needed to ensure patient safety when urgent medical attention is needed. Patient tags with RFID chips meet this need.

1.2.4 Electronic Payment and Toll Collection

There has been a lot of effort to automate the payment by using electronic payment devices like smart cards, etc. One of the first successful implementation of such an electronic payment using RFID was done by Mobil Oil Corp. and is known as SpeedPass [25].

Speed Pass is a key chain RFID device first introduced in 1997 [11]. This device is used for payment at various Mobil and Esso gas stations. It has also been deployed at various fast-food restaurants and select supermarkets. This device provides a lot of convenience to the consumers as they do not have to stop for long to make payment and wait to get the change and collect the receipt. As these devices are contact less, it does not require the costumer to even roll down their windows.

In most countries, the highway authorities collect some toll for the highway usage. To collect these tolls, toll booths are created at various entry and exit points of the highway. In order to pay the toll, the vehicles have to be stop, roll down the car window, take out the money and hand it to the toll-collector, get the change and receipt, roll up the window, and start driving again. This whole process slows down the journey and normally leads to a traffic bottlenecks at these toll booths. This may also lead to loss in the toll collection, as some users may opt for alternate routes. Automatic toll collection helps in such scenarios by reducing the queue length and the time spent at the toll booths. EZ-pass[7] is one such successful implementation of automated toll collection which uses RFID tags on the cars and other vehicles.

Various other applications have also been built over RFID based systems. These include animal tracking [28], asset management [17], article tracking [21], etc.For the development of RFID applications, some effort has been made to provide application frameworks. These application frameworks enable the building of RFID applications effectively and easily. However most application frameworks have not been generic in nature and have stuck to certain predefined interfaces. We discuss two of these application frameworks, the EPC Global Architecture Framework and ALE Application Framework.

1.2.5 EPC Global Architecture Framework

EPCglobal [6] envisions a network of EPC [29]-enabled data services that is used by trading partners to enable near-real-time tracking information on items in their supply chains. This vision is termed the EPCglobal Network. The EPCglobal Network [31] introduces a few dedicated components, such as the Object Naming Service (ONS) [27] and the EPC Information Services [26](EPCIS), that may or may not be needed for an application. However, chances are that one will end up using or developing similar components to meet ones requirements.

There are three broad activities accomplished by the EPCglobal architecture, each supported by a group of standards within the Framework.

EPC Physical Object Exchange

Subscribers of this service exchange physical objects that are identified with Electronic Product Codes (EPCs). For many end users of the EPCglobal Network, the physical objects are trade goods, the subscribers are parties in a supply chain for those goods, and physical object exchange consists of such operations as shipping, receiving, and so on. There are many other uses, like library or asset management applications that differ from this trade goods model, but still involve the tagging of objects. The EPCglobal Architecture Framework defines EPC physical object exchange standards, designed to ensure that when one subscriber delivers a physical object to another subscriber, the latter will be able to determine the EPC of the physical object and interpret it properly.

EPC Data Exchange

Subscribers benefit from the EPCglobal Network by exchanging data with each other, increasing the visibility with respect to the movement of physical objects. The EPCglobal Architecture Framework defines EPC data exchange standards, which provide a means for subscribers to share data about EPCs within defined user groups or with the general public.

EPC Infrastructure

In order to share EPC data, each subscriber carries out operations within the area of its physical center. It creates EPCs for new objects, follows the movements of objects by sensing their EPC codes, and gathers that information within the organization. The EPCglobal Architecture Framework defines interface standards for the major infrastructure components required to gather and record EPC data, thus allowing subscribers to build their internal systems using interoperable components.

In addition to tagging objects with RFID tags and installing readers to record the inventory, two things are needed to make it useful for the tracking at various points in the supply chain – a service that maps EPC codes to the relevant product information, and a service that can provide the address of a particular EPC code's information look up service. EPCglobal calls these services the EPC Information Services and the Object

Naming Service, respectively.

Object Naming Service (ONS)

The ONS is an EPC resolution service that provides the means to look up a service resource that provides further information about the item identified by a particular EPC. The ONS uses the standard Domain Name Service (DNS) for resolving EPCs. The EPC query and response formats therefore use the DNS standard.

EPC Information Services (EPCIS)

The EPCIS is an EPCglobal standard which permit applications to share and use EPC data across enterprises. EPC related data are captured using the EPC capture interface, and are queried using the EPC Query interface, as shown in figure 1.1. Additionally, the EPCIS provides a common model for location information and other important data.

The EPCIS focuses only on the service interface and semantics of EPC-related data, such as the location information. Vendors extend the EPCIS services with add-on functionality.

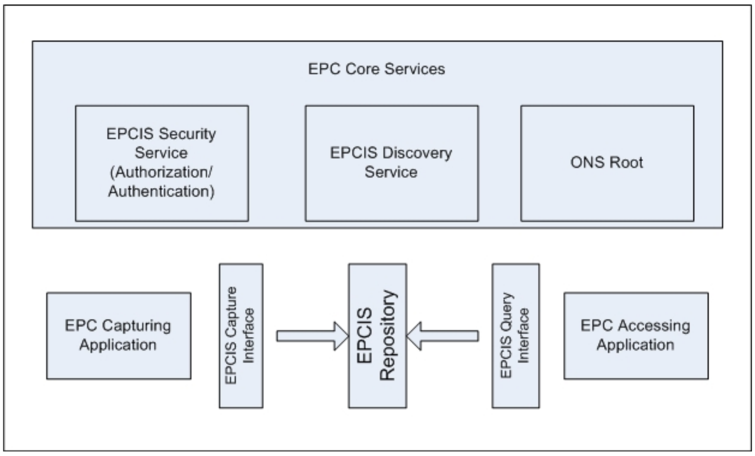

Figure 1.1: EPCIS Components

1.2.6 ALE Application Framework (AAF)

AAF [23] is a framework developed by Kyuhee An and Mokdong Chung at the Department of Computer Engineering, Pukyong National University, Korea for RFID application development. This framework allows the clients to develop and use the RFID applications based on ALE [39] efficiently and easily. The role of the ALE is to provide means to process the event data which have been collected by the RFID reader and to deliver them to the higher-level applications. An RFID reader delivers identified tag data to the middleware, ALE Engine. Middleware filters out various redundant tag data, and transmit accumulated/filtered tag data index to EPCIS [26] or to applications.

The various modules of this framework include the following.

Data Manager Module

Data Manager offers functions to access outer systems through databases and web services. It includes a data access module, which is in charge of input and output processing of the data by approaching the database.

Security Manager Module

Security Manager provides an authentication mechanism to the entire platform for the distributed service environment connected to the network. This module uses Kerberos [34], which is an authentication protocol based on the symmetric-key and the concept of Single Sign-On [10].

Event Manager Module

Event Manager implements the function of defining and processing the EPC event. It includes mechanism by which an EPC event may be defined as desired by a client along with the event handler to process the event received from the middleware.

Business Process Module

Business Process Manager implements the business logic in the applications using the sequence of the events generated by the Event Manager.

1.3 Organization of Report

The the rest of the report is organized as follows. In chapter 2, we describe the background of RFID technology and various RFID system components. In chapter 3 we give a detailed architectural description of the Application framework. It also traces the implementation details including the communication architecture, initial configurations and the various system requirements. We describe in chapter 4 two applications built using our Application Framework – Electronic File Tracking and Postal Bag Tracking System and finally conclude this report with chapter 5.

Chapter 2

RFID System Background

Radio frequency identification (RFID) is a term that describes a system which is used to tracks objects, where the identity (unique serial number) and data of the object is transmitted wirelessly using radio waves. Data is stored on an electronic data-carrying device (tag), which is generally attached to the object being tracked. The power supply to the tag and the data exchange between the tag and the reader are achieved without having any physical contacts with the reader devices, rather using alternating magnetic or electromagnetic fields for the same.

2.1 Components of an RFID System

The various components that comprise an RFID system are RFID hardware comprising of RFID tags and RFID reader; a data processing interface software called RFID middleware; and software applications (figure 2.1).

2.1.1 RFID Tag – The Transponder

An RFID tag is composed of an antenna, memory, an encapsulating material and optionally a small processing unit. These tags can be categorized on the basis of many characteristics.

One of the characteristics is the way in which tags are packaged. They may be housed in PVC buttons, glass vials, paper labels, or plastic cards.

The tags can be either active or passive. While the active tags have an on-chip power source, passive tags use the power induced by the magnetic field of the RFID reader. There

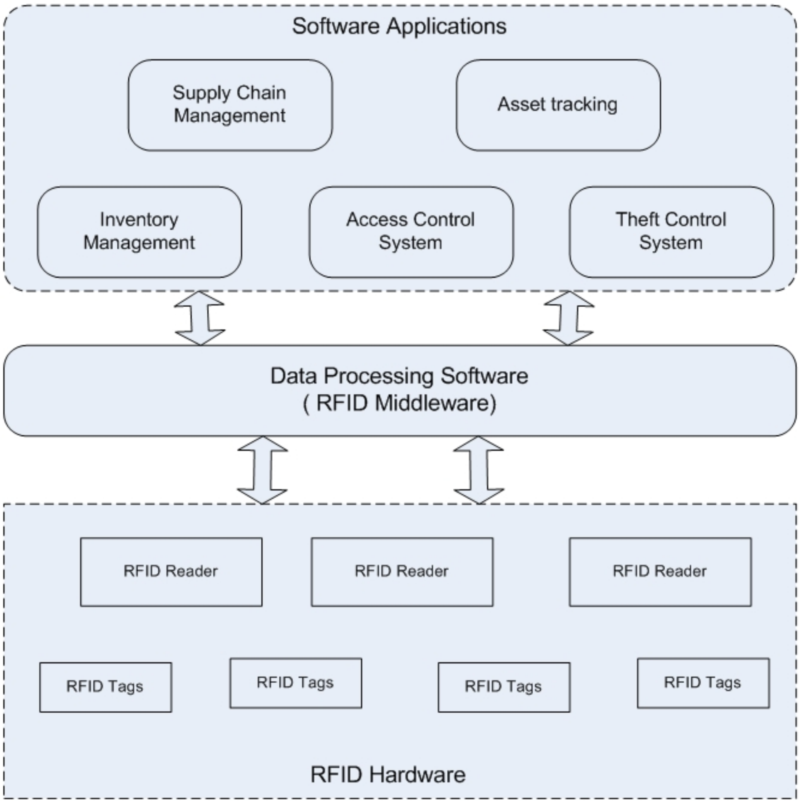

Figure 2.1: RFID System Components

is a third type of tag, called "semi-active tag", which powers its own communications and may even be capable of communicating directly with other tags without a reader. The passive tags are cheaper but have lower range ($<$ 10 meters), as compared to the active tags.

Tags come with varying amounts of storage capacities [1]. The storage capacity of tags ranges from the 1-bit – used for theft prevention, to a few kilobytes, used in short range access control management systems. The tag memory can be either read-only, write-once or rewritable. The read-only tag memory is set to a particular value at the manufacturing time. On the other hand, the value can be set once in a write-once tag memory. After setting the value, these tags act like read-only tags. Write-once types of tags are generally used to track the objects which have to travel across multiple organizations and the tag information needs to remain the same. The tag with rewritable memory are used in the applications where the tag data needs to be altered during the course of its usage. These types of tags can be reused with different objects. However, this type of tags are the costlier ones as compared to the other two.

2.1.2 RFID Reader – The Interrogator

RFID readers, also called interrogators, are used to recognize the presence of RFID tags in the range and communicate with it. This communication is done by transmitting and receiving RF energy. An antenna of a nearby tag picks up this energy, and then converts it into electrical energy via induction. This electrical energy powers the semiconductor chip attached to the tag antenna, that stores the tag's identity and other data. In order to communicate with the tags, an RFID reader needs to have one or more antennae. Some readers have only one antenna, while other readers are able to support many antennae which can be placed at different locations. The limitation on the number of antennae, a reader can control, is the signal loss on the cable connecting the transmitter and receiver in the reader to the antennae. Some readers use different antennae to transmit and receive, whereas others transmit and receive the RF waves using the same antenna. Readers communicate with the network and other devices through a variety of interfaces. Historically, most RFID readers have had serial interfaces using RS-232 [19] or Ethernet interface using RJ-45 [9]. However, some new generation mobile readers also have Bluetooth or wireless

Ethernet communication interfaces. The RFID readers are available in different frequency ranges – LF, HF, UHF, etc. The operating frequency range of the LF readers is less than 135 KHz. The ISO standards are ISO 11784 [12] and ISO 11785 [13] define the LF based RFID infrastructure. The LF systems are used in tracking of animals. The HF readers operate at 6.78 MHz, 13.56 MHz, 27.125 MHz, 40.680 MHz, etc. Some standards defined in this range are ISO 14443 [14]and ISO 15693 [15]. The HF based systems are used in applications involving ID and access control management. The UHF readers operate in the range of 300 MHz to 3 GHz. The important frequencies in this range are 433.920 MHz, 869 MHz and 915 MHz. The standards defined for UHF based RFID tags in this range include ISO 18000 and EPC Gen2. The UHF based RFID systems are used in application involving object tracking, and asset and inventory management.

The RFID readers are usually capable of reading multiple tags simultaneously in the range. They apply various singulation and anti-collision algorithm to accomplish this task. The various anti-collision algorithms used by the readers includes Aloha, Slotted Aloha, binary search based enumeration [30], etc. These readers also have either a micro-controller or a micro-computer to implement the communication and anti-collision protocols. Some readers even have a web server hosted – a mechanism which provides an easy to use interface for various reader operations.

The readers have different read ranges depending upon the operating frequency and power level. The read range varies from a few centimeters to few meters. Some readers also have memory buffers to hold the data read from various tags seen. This facility of buffers allows to retrieve the tag identification data even when the connection with the network is lost. Some readers do not have such in-built buffers. They just read the data and put it on its interface for the requesting party to receive.

2.1.3 RFID Software Applications

Applications are the software components of the system. These applications require the data to be collected from the RFID hardware and middleware. The applications then apply various business rules on this data to extract information. The business rules vary in complexity depending upon the application's requirement.

The applications interface with the readers using the proprietary APIs provided by the reader manufacturer. There has not been any standardization on the reader API front, making it necessary for the application developer to know about the various reader APIs available and to change the application code in order to make the application use the services of other type of reader.

Each of the connected readers keep reading all tags in the read range. This can result in thousands of RFID read observations per second including the RFID tags being read multiple number of times. In addition to the sheer volume of data, the raw observations need further processing to be meaningful for the enterprise applications.

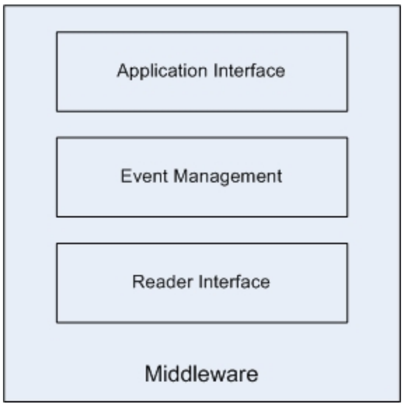

Figure 2.2: Components of RFID Middleware

2.1.4 RFID Middleware

An application when interfaced directly with the RFID reader, has to process humongous amount of data. Such applications also have to take care of different connection-interface used by the reader. Handling these different interfaces becomes a huge task as interfaces require different APIs to program.

In order to make applications independent of various types of readers and their connection interfaces, there is a need of an intermediate layer between the RFID reader and the application. This requirement is fulfilled by a software unit (layer) called the RFID middleware.

The middleware provides the following facilities.

- A device characteristic independent interface to the application.
- Processing of the raw data and reporting only the aggregated and meaningful data as configured by the application.
- Providing an application-level interface for managing readers and querying RFID observations.

The RFID middleware incorporates several components (figure 2.2) as described here.

Reader Interface

The applications can interface with readers connected to the system by making use of the various APIs provided by the reader. This however needs huge amount of efforts of application porting. The reader interface component of the middleware provides the means to eliminate this effort by exposing a single abstract interface to the applications.

Event Management

A typical RFID-enabled environment has several readers employed for identifying the tags to track objects. Each of these readers transmit RF signals several times a second in order to read the RFID tags around them. This can result in thousands of RFID tags being read per second. Exposing raw observations from the readers to applications will require enormous processing at the applications end. In addition to the sheer volume of data, the raw observations need further processing to aggregate data and present only the meaningful data to the enterprise applications. As the RFID technology is still not immune to data losses, it is possible that in some cycle a tag is identified whereas in other cycles, it may not be identified. With raw processing at the applications, the application will have to continuously adjust to the fluctuating observations coming from the readers.

The raw observations from RFID readers therefore lack the meaningfulness for the applications. More processing needs to be done to map these raw observations to coarser events that are meaningful to applications. A middleware helps by consolidating, aggregating, and filtering the raw observations coming from readers and sensors and provides application-level context. The process of smoothing out the raw RFID observations coming from readers to make them more meaningful for enterprise applications is called event filtering. This functionality is provided by the event manager component as shown in figure 2.2.

Application Interface

The applications require a service-oriented interface that provides application-level semantics to the collection of RFID data. Following the principles of service-oriented architecture, this interface has to be loosely coupled and asynchronous.

2.2 Standards

The tags are used for tracking the objects not only within the organization but also across organizations. Therefore, the tags have to be read at different places and keeping same type of reader at all the place is not possible. In order to read the tags at different places and to enable the reader to communicate with different tags, there is a need of standardization. These standards define, among other things, the various packet formats for communication, transmission protocols, the initialization, the singulation and the anti-collision mechanisms.

There are several standards defined by the ISO. The ISO is the worldwide union of national standardization institutions, such as BIS [4] (India), DIN [5] (Germany) and ANSI [2] (USA).

There are different standards based on operating frequency ranges and on the application for which the RFID system is intended for. Some relevant standards are the following.

1. ISO 14443 [14]: Proximity coupling smart cards operating at 13.56 MHz, and offer maximum read distance of 10cm.

2. ISO 15693 [15]: Vicinity coupling smart cards operating at the 13.56 MHz frequency, offering maximum read distance of 1 meter.

3. ISO 18000 [16]: RFID for item management: Air Interface operating at different frequency ranges.

4. EPC Gen2 [29]: UHF RFID protocol for communications at 860 MHz – 960 MHz.

The RFID system, as described in this chapter, has several variables like – interfacing protocols, different reader connection-interfaces, different tag types and memory organization. These different features result in difficulties to make and to port applications to multiple readers in an enterprise usage. The RFID middleware software component in the system allows to overcome these variations and difficulties by giving an abstract view of the RFID hardware to make the development of RFID enterprise applications easier.

Chapter 3

System Design

The design of our system has a three-tier architecture (figure 3.1) with application framework, middleware (SmartRF) and the RFID hardware being its components. The RFID hardware consists of the RFID readers and the tags. SmartRF is a software component which provides a device independent interface to the applications for accessing the RFID hardware. The application framework uses the services of the middleware to present a generic application building environment. This framework uses a device independent visualization of the RFID hardware provided by SmartRF to build the applications. The RFID applications are independent software which use the services of the Application framework for accessing the tag data read through the hardware to implement the business logic. These applications are developed using a generic application framework.

3.1 SmartRF – The Middleware

SmartRF is a middleware developed at the department of CSE, IIT Kanpur. It provides a device neutral, protocol and platform independent interface to the application framework. It provides the following functionality to the application framework.

- It is responsible for interaction with the hardware. It provides access to the devices and tags through tag and reader abstraction layers in a manner independent of their specific characteristics .
- It provides the application framework with the tag data from various reader devices.

It reports the data after applying application-specific filtering and after aggregation of the raw data.

- It gives a set of functions to the applications to interface with the RFID hardware, in a device neutral manner.

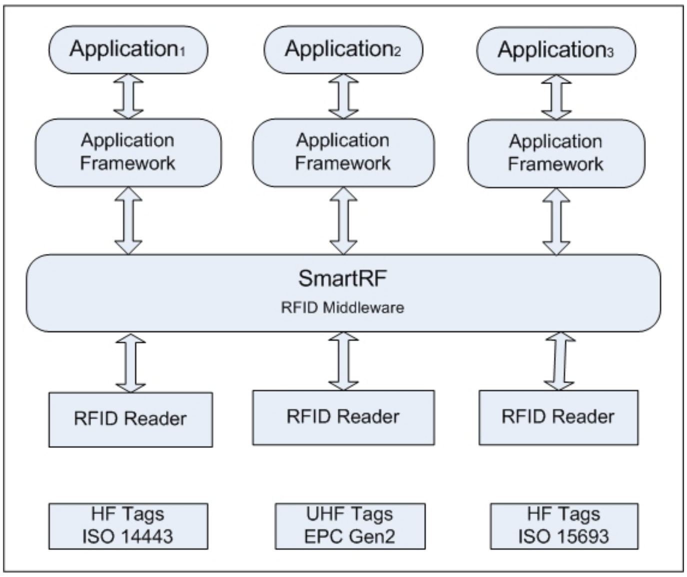

Figure 3.1: System Architecture

3.2 Application View of the Hardware

The RFID hardware is presented to the application as an abstraction of data-stream.

3.2.1 Data Streams

An RFID reader generally has multiple antennae connected to it. These antennae may be placed in groups at different locations for tracking purposes. A group of antennae at one tracking point may belong to single or multiple RFID readers. There may also be readers whose antennae are part of various groups allowing a single reader to be a part of multiple tracking points. This functionality is provided by introducing the notion of channels which allows to combine multiple reader-antenna pairs as a single entity called channel. The channel is a virtualization of a tracking point and is used to associate multiple reader-antenna pairs to the tracking point.

A tag might be seen by one or more of the antennae connected to the RFID readers, depending upon the air interface and tag illumination. SmartRF, therefore, uses a concept of data stream to represent a reader-antenna pair. The number of data streams for a reader is equal to the number of antennae connected to it. Thus, every data stream acts as an independent source of data to the application, received by the middleware. The data streams are exposed to the applications, which can configure the data streams in a group known as channel according to their specific requirements. For example, let's consider a system with two readers R_0 and R_1 with antennas A_{00}, A_{01}, A_{02}, A_{03} and A_{10}, A_{11}, A_{12}, A_{13} respectively. The applications will therefore be provided with the following data streams (R_0, A_{00}), (R_0, A_{01}), (R_0, A_{02}), (R_0, A_{03}), (R_1, A_{10}), (R_1, A_{11}), (R_1, A_{12}) and (R_1, A_{13}), as shown in figure 3.2.

In most available readers, antennae can transmit as well as receive the RF signals. However, there are some readers which use different antennae for transmission and reception. This property of the readers puts a restriction on the usage of each single antenna as an independent data source. This limitation is incorporated by introducing the concept of 'association'. Here, the administrator specifies the antennae which are associated through a one-time system wide configuration done at the installation time. The data streams corresponding to these associated antennae are dependent on each other. While making use of a data stream, all other data streams in the association are automatically marked non-available during channel creation as explained in the next section.

3.2.2 Channels

Channel is a logical grouping of different data streams. Applications define channels by grouping data streams as per their requirements and physical deployment of readers and antennae. The channels are used to logically represent an RFID gate that incorporates one or more RFID reader-antenna pairs. For example, if a door has two antennae placed orthogonally, the channel will be defined to incorporate both of these data streams. The multiple data streams in a channel may belong to same or different readers. Once a data stream is selected as a part of a channel, it cannot be used further by other channels, until the channel is destroyed and the associated data streams are freed. An application can create any number of channels as long as there are data streams available.

During the creation of a channel, if an antenna of the selected data stream is a part of an association, then the other data streams consisting of the corresponding associated antennae are automatically pulled in as a part of that channel.

An example of channel creation by the grouping of data streams is given in figure 3.3. In this figure, R_1 and R_2 are the readers with their respective antennae denoted as A_{10}, A_{11}, A_{12}, A_{13} and A_{20}, A_{21}, A_{22}, A_{23}. In this example, four channels C1, C2, C3, C4 are created here. Channel C1 and C3 have two data streams each, whereas channel C2 and C4 have one data stream each. However, in the figure, the channel C2 is shown with two data streams. This happens because of association. Here, antennae A_{12} and A_{13} are part of an association. Thus, when the data streams having either A_{12} or A_{13} as their antennae is selected as part of the channel, the other data stream with associated antenna is automatically selected as part of the channel.

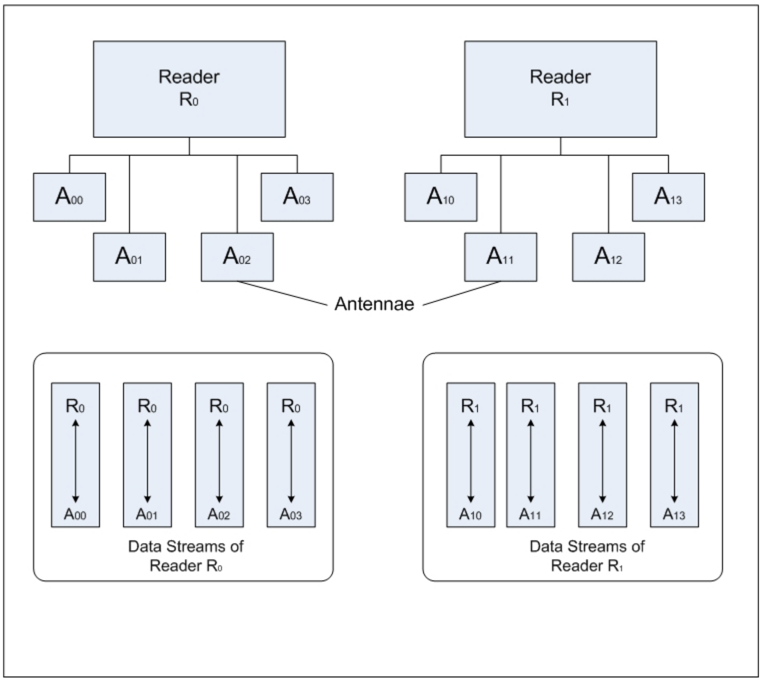

Figure 3.2: Readers and Data Streams

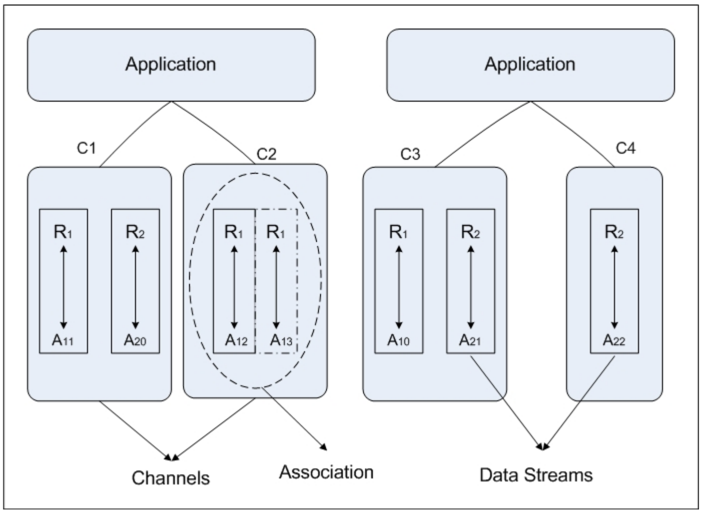

Figure 3.3: Channels– Grouping of Data Streams

3.3 Application Framework

The application framework defines a generic way to build RFID applications. These applications can be categorized based on their requirements.

The set of applications based on asset tracking, movement tracking and supply chain management require logic built in the application design. These kind of applications identify items based on certain user-defined events.

The other set of applications like item theft management, access control do not require significant processing or analysis of the tag data, as they only log the tag-ID read by the RFID reader.

There is another set of applications, that not only log and process the tag data, but also update it.

The framework described here grants ability to develop applications catering these kind of requirements with the help of SmartRF as the underlying middleware. The logic involved in complex analysis of the data can be easily provided using this framework. The framework builds over the concept of data streams and channels to define application specific events. For example, in an application to track the items entering into a warehouse, 'Entry to the warehouse' is an event which signifies the entry of the item into the warehouse. These events are characterized by regular expressions where the alphabet denotes the channels. The string accepted by this regular expression specifies that a tag data has been read in the channel order as specified in the expression. The acceptance of the string signifies occurrence of the event, defined by the regular expression. Consider an example, where we have a system with channels C_1, C_2, C_3, C_4 and C_5 as configured by the application. Events E_1 and E_2 are defined as the following.

$$E_1 \rightarrow (C_1|C_2)C_3^*$$

$$E_2 \rightarrow C_4^+C_5$$

The event E_1 is reported whenever a tag is read by channel C_1 or C_2 followed by multiple or no reads by channel C_3. Similarly the event E_2, is reported when a tag is read one or more times by channel C_4, followed once by a read from channel C_5. A capability to define events in such a format allows an application developer to infuse intelligence into the application and thus build complex applications with little effort.

RFID hardware does not guarantee 100% reliability in reading of tags. Many a times this leads to read misses by channels and thus, affecting the robustness and reliability of the system. This shortcoming is handled by the framework using the event specification defined by regular expressions. This improves the performance and efficiency of the system as it can accept strings having read misses by some channels. For example, the application framework may report event E_1 followed by E_2 even when the tag is seen by C_1, C_4 and C_5 in that order.

3.4 Implementation

The application framework interfaces with the middleware to collect data, read by the RFID readers. The framework also has to provide data to the application to process and present it to the end-user. The application framework takes the configuration parameters specific to the application and passes it on to the middleware. There are various functionalities provided by the middleware which are utilized by the framework to accomplish these tasks.

In the following sections, we explain the implementation of these various aspects.

3.4.1 Interfacing with the Middleware

The middleware operates as a server daemon which services the various application requests using client-server paradigm. There is a set of APIs provided by the middleware which internally handle the client-server communication. These communications are accomplished using sockets (Winsock for Windows and UNIX sockets on UNIX-based systems).

Each instance of the application framework acts as a client to the middleware server. There are various API functions utilized in interfacing the application framework with the middleware. Some of them are explained below.

Reader List

This functionality of middleware provides the list of readers currently connected and accessible to the system running the middleware daemon. It also provides various parameters and other information about the readers. These include the user defined name of the reader, the manufacturer name, the number of antennae attached with each reader and their corresponding user-defined names which are used for the identification of individual antenna.

The user-defined names are provided by the user to the middleware using a configuration file. The names of the reader can be defined in such a way that it reflects the physical location or identification of the reader. The same is also the purpose of the naming of the antennae.

This list is presented to the user, on the client side, to select antennae and readers according to their requirement to create channels (as explained in the section 3.2.2).

Creating and Deleting a Channel

This function is used by the framework to provide the middleware with the information about various data-streams (section 3.2.1) selected to create a channel. It provides a channel identifier which is used by the application framework to perform operations on the channel. These operations include writing and reading the tags in the read range of the antennae of the data-streams which are part of the channel.

There are various parameters related to the channels which can be configured using the application framework. One of the most important parameters is the specification of the duplicate time. This time specified in seconds, configures the duration during which a tag data is reported only once, whatever number of times it might have been seen. Another important parameter related to the channels is the filter. Here, the user can specify a string of bytes including some wild cards to filter out the unwanted tags.

Reading from the Channel

After creation of a channel, the application can use the APIs to receive data from the middleware. The data is received after processing for the duplicates and filtering, based on the specified duplicate time and the filter string. In case when the channel has no data, the middleware can behave in one of the two ways, which can be specified through the function argument. One way is where the middleware performs a non-blocking read and responds with the read data length to be zero. Another way is where the middleware waits and responds only when it has data for the channel. The second one is called the blocking read. In the case of blocking read, a GUI application may become non-responsive for the blocking duration. This is handled by the framework by spawning a thread which waits on the blocking call. This mechanism allows multiple read calls to be made for different channels at one time.

There is another case, where the number of bytes available in a particular tag is less than the specified value in the read API call. In that case, the middleware responds with the available amount of data and its length in bytes.

3.4.2 Interfacing with the User-Applications

The data received by the application framework is made available to the user-applications through a database server. All data which is received by the framework is stored in this database. The format used to store the data is the following.

$$< Data, Channel\,Name, Time >$$

The data stored by the application framework is the raw data, specifying only the "seen" information of the tag. This information includes the time and channel where the tag was seen. The user application has to make sense out of this data and present it as information to the end user. In order to accomplish this task, some more information is required. This is achieved by specification of events (as explained in section 3.3). The specifications are in the form of a regular expressions which are stored in another database table in the following format.

$$< Event\,Name, Channel\,Sequence >$$

The application framework is provided with a graphical user interface, which makes it easier for the application developer to perform various operations. The GUI provides the following interface.

Creation of Channel

A list of readers, along with their connected antennae, is presented to the user. The user (application administrator) can chose antennae from this list to be grouped together to form a channel. Any number of antennae can be chosen provided they have not being already selected by some other channel.

Configuring Channel Parameters

Various channels created by the user can be configured using the application framework. The important parameters that can be configured include the duplicate time and the filter string. The filter string is entered in form of a hexadecimal string where the number of alphabets is equal to twice the number of bytes specified to be read. We can use wild cards such as 'x' or '*' for a "Don't Care" hex-digit, which is not to be considered during the filtering.

As an example consider a tracking application which uses an encoding where the first three bytes denote the destination code. If, at one location, we want to track only those assets which have to be dispatched to the destination having the 3 byte code 11662A (in hexadecimal), the filter that can be specified is 11 66 2A xx xx xx xx xx xx xx xx.

Parameter	Type	Description
Number of Bytes	Number	Maximum number of bytes which to be read
Duplicate Time	Seconds	Duplicate window time in seconds
Filter Value	String	Filter value for the channel filter

Table 3.1: Channel Parameters

The other parameter is the duplicate time specified in seconds. As an example, consider the case of a channel created by grouping antennae at the gate of a warehouse. It is expected that during a normal operating conditions, the asset to be tracked takes about one minute to cross the gate. During this one minute, the channel will have hundreds of reads for the same asset as the tag will be read multiple times. If this information is presented to the application without aggregation it causes an enormous computation load. In this case, if we specify the duplicate time of about 60-70 seconds, all the reads of the same tag will be reported as one read.

The layered design of the system allows modifying the implementation details easier and without bothering about the implementations in other layers. Various parameters are available which allows the user to customize the framework client according to his requirement. An easy to use GUI is provided which makes specification of the parameters easier and faster. The server-client implementation of the middleware-application interface allows many framework clients, at different places on the network, to use and share the services of the hardware in a distributed environment. The usage of the database server to consolidate all the data allows better resources to be dedicated for data handling. It is also advantageous as it allows data to be available on the network.

Chapter 4

Applications

We have used the application framework, described in the previous chapter, to build two applications for electronic file tracking and postal article tracking. In the development of these applications, we had to consider only the processing of the data, whereas the data handling from the middleware was handled by the framework. The processing of data involved application of the logic rules defined as various events on the data.

4.1 Electronic File Tracking system

4.1.1 Current System of File Tracking

In most offices, paper/plastic files and folders are used to keep records and to communicate among various departments. For example, consider an office scenario having various departments spanning over same or different locations. In order to communicate information from one department to another, or within the same department, files are used to organize the information. The files are then carried between the departments by a person, a courier or mail service. When carried by a person, a log book is used to record the arrival or dispatch of files through various departments. The entry in the log book is made by the concerned person receiving the file. This process of manual entry into the log book is an error prone and cumbersome procedure. It may happen at times, that the person fails to carry the log book or an entry is missed from being recorded due to the temporary unavailability of the concerned person. The tracking of files in such cases becomes difficult, as file entries in the log book have been unintentionally missed.

Efforts to automate this process using barcodes [3] have not been very effective. The primary reason for this is the involvement of a manual process of making an entry of the file using the barcode scanner. The only advantage of such a system is that the data for tracking is available and consolidated at a central database which can be queried electronically.

4.1.2 File Tracking using RFID

RFID systems, having the ability of automatic identification, has the potential to improve upon the current file tracking by reducing the manual errors of the current system. It also consolidates the RFID data acquired at various tracking points at a central database to facilitate the querying of tracking information about a file. We have developed an RFID based electronic file tracking system and used it in our experimentation. This system uses SmartRF (section 3.1) as the middleware and the tracking logic is specified using our application framework. The event logic specified through the framework is stored as regular expressions in a database. This is then used by the application to figure out the occurrence of events. RFID tags, first enclosed in a plastic package before they are placed on the file surface (figure 4.1), uniquely identifying each file. The plastic packaging serves two purposes. Firstly, it provides protection against physical wear and tear, secondly, it provides some air-gap between the tag and the file surface facilitating better inductive coupling. Reader antennae are placed at each of the tracking locations to track the file movement. In our implementations, the reader and antennae are placed at the entrance and exit doors of the department. The configuration of the hardware and software varies depending upon the number of entrances for a particular location. In places with a single entrance, a reader is used with four antennae. The single entrance acts as both entry and exit point for the department. In places where the entrance and exits are different, even two antennae are sufficient depending upon the width of the entrance and exit channel. The antenna placement in case of single entry and exit door is shown in figure 4.2. The antenna placement and orientations are such that, two antennae always read a tag earlier than the other two during entry or exit of the file. SmartRF is configured such that the two antennae which read the tags earlier form one channel and the other two antennae

Figure 4.1: RFID Tag Packaging and Placement on the File

formed the other channel. For example, the reader R0 and its antennae A0, A1, A2 and A3 are arranged in such that antennae A_0 and A_1 read the tag earlier than antennae A_2 and A_3 during the entry and vice-versa at the exit, as shown in figure 4.3.

The application framework is used to create two channels namely, C_1 and C_2. C_1 is created with data streams R0-A0, R0-A1 and C_2 with data streams R0-A2 and R0-A3. The entry and exit events for this location were specified as the following.

$$Entry \rightarrow C_1^*C_2$$

$$Exit \rightarrow C_2^*C_1$$

Thus if the file is seen at C_1 multiple times, an aggregation takes place which reports only *Entry* to the application. The event Entry is recorded if a tag is seen at C_2 after being seen at C_1 one or more times. Similarly, the event Exit is recorded when the file data is read at C_1 after one or more read by C_2.

4.1.3 Implementation of the System

A web based application is developed for tracking the file. During the creation of the tag, the file name and its corresponding information (creating department, name of the owner, date and time of creation, etc.) is entered through the web application and stored in the database. All tag data is then written to the central database. On a query for a particular file, the application displays the list of events for that file (figure 4.4 and figure 4.5). The components involved in the development of this system are the following.

RFID Hardware – EPC Gen2 Tags (96 bit UID), UHF RFID Readers.

Software – SmartRF – The RFID middleware, Generic Application Framework, Central Database implemented using MySQL Community Server, Apache HTTP Web Server.

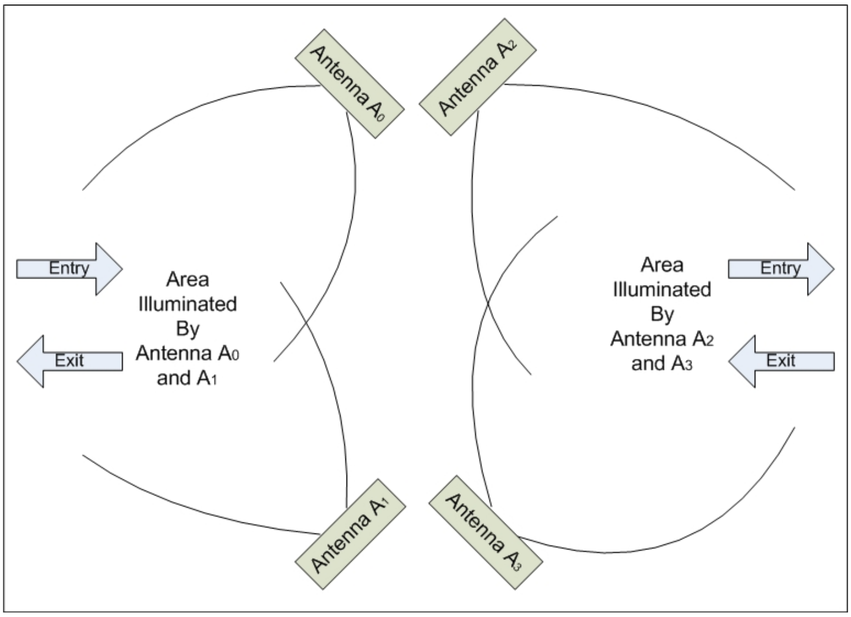

Figure 4.2: Antenna Placement and Orientation at the Room Entrance

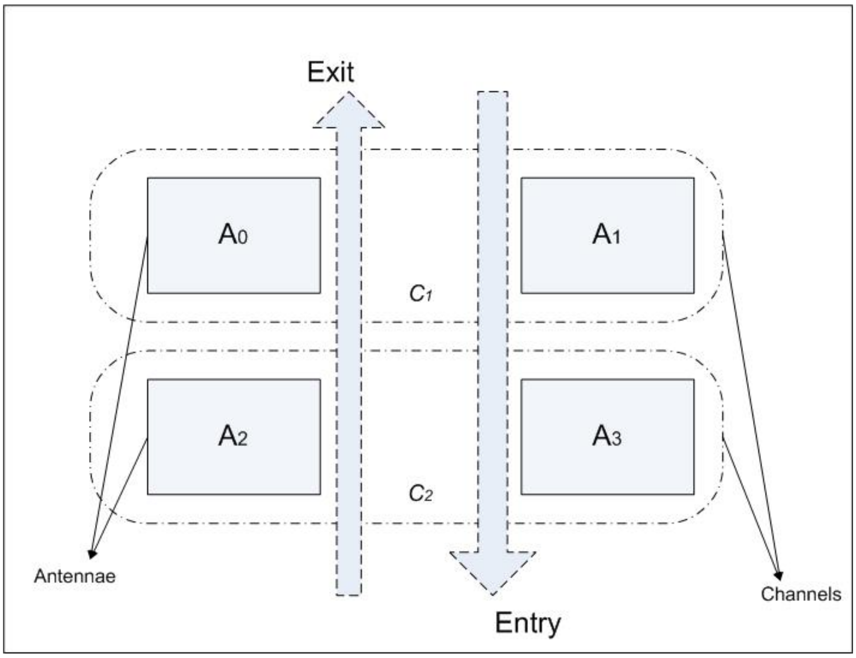

Figure 4.3: Entry and Exit for a Room

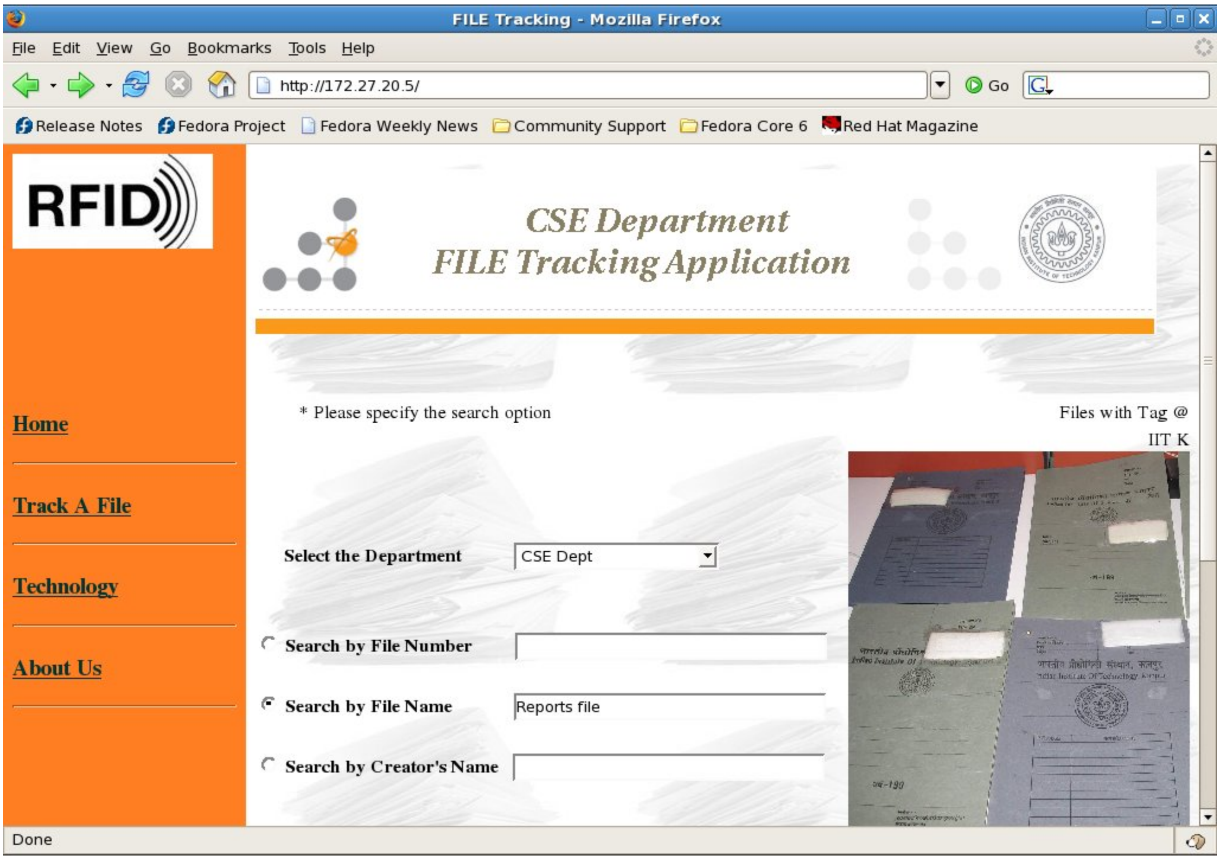

Figure 4.4: Portal for Electronic File Tracking (I)

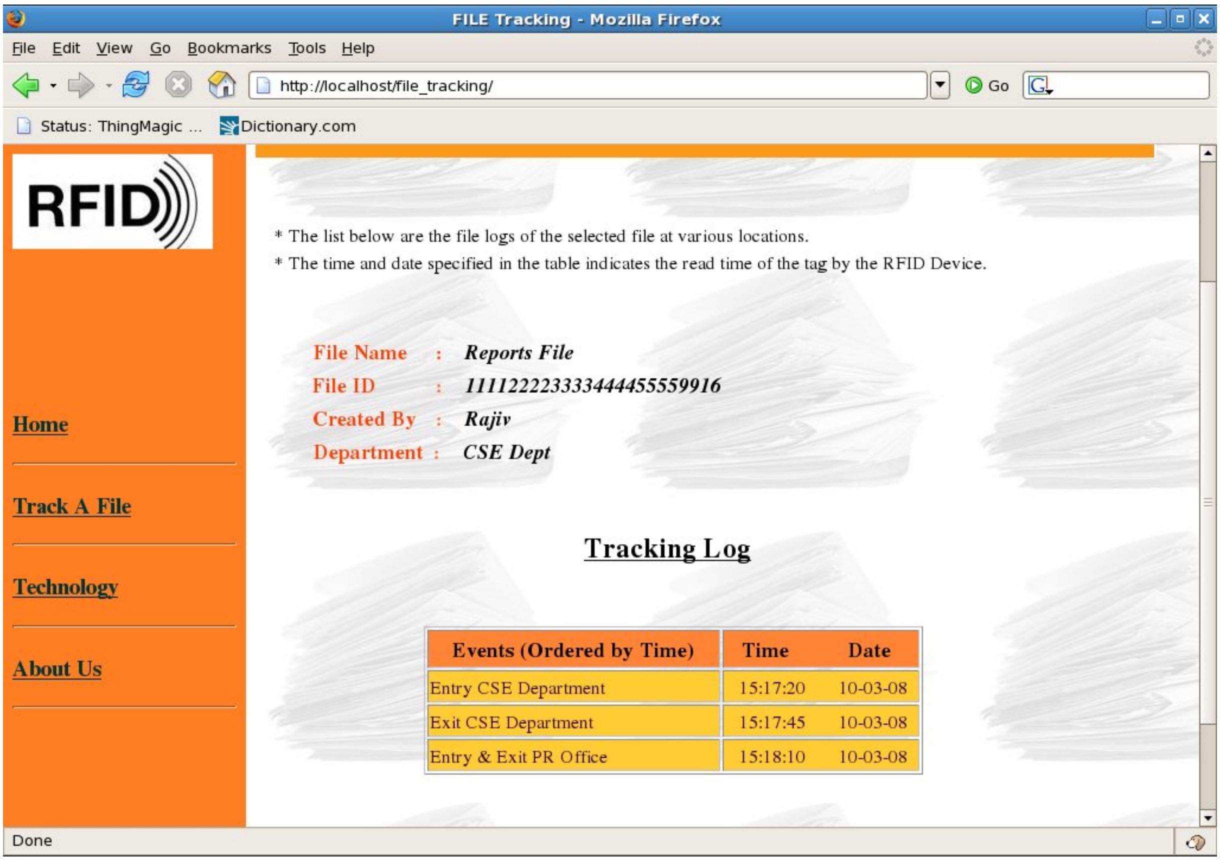

Figure 4.5: Portal for Electronic File Tracking (II)

4.1.4 Experiments and Results

The system was evaluated over a period of one month with over 100 observations spanning throughout the experiment. The tracking percentage without using regular expressions for event specifications was 70 - 100%. An important observation made during the testing of the system was that the read misses by the readers were quite significant. The read percentage dropped significantly with the increase in the number of files being carried together, as shown in figure 4.6. The regular expression specifications provided by the application framework was very useful and effective in capturing the events involving missed reads. The performance of the system significantly improved by including this feature. The evaluation involved testing of the system with different tags, readers, antenna placement and orientations, number of files carried together, packaging materials, tag placements and other environmental conditions.

Tags with various antenna designs were used. The performance of two of the tags from two different manufacturers is provided in figure 4.6. In another experiment, we placed the tags on the files in a staggered manner. Tags were not placed at the fixed position on the files. The performance in the percentage of reads between staggered and non-staggered tags is shown in figure 4.7. The files with staggered tags had a better performance due to the absence of de-tuning factor between the tags as opposed to the case of non-staggered tags. The packaging materials used for tags, played a significant role. Thicker packaging materials provided better results compared to thinner ones or the ones with no packaging material. This was due to the fact that, thicker materials provided the tags with an larger air gap and thus improved their performance. The experiment results were not very different when carried out with different UHF RFID readers.

4.2 Postal Article Tracking

4.2.1 Current System of Postal Article Tracking

The Department of Post is an organization under the Ministry of Communication and Technology, India. One of the postal services called the Speed Post, links more than 1200 towns in India. The number of articles using this postal service are more than 10 million

[37] monthly. The current operation of this postal service takes place in the following manner.

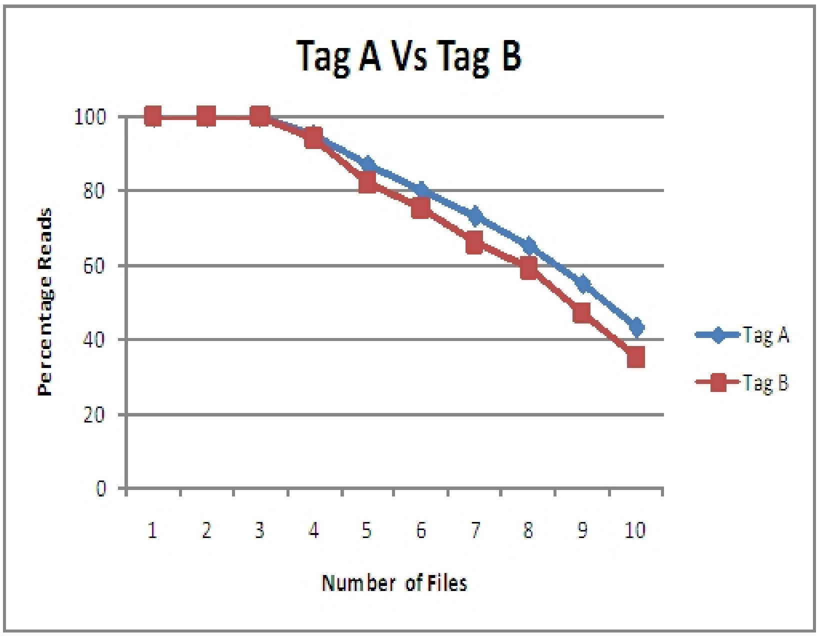

Figure 4.6: Comparison between Tag A and Tag B

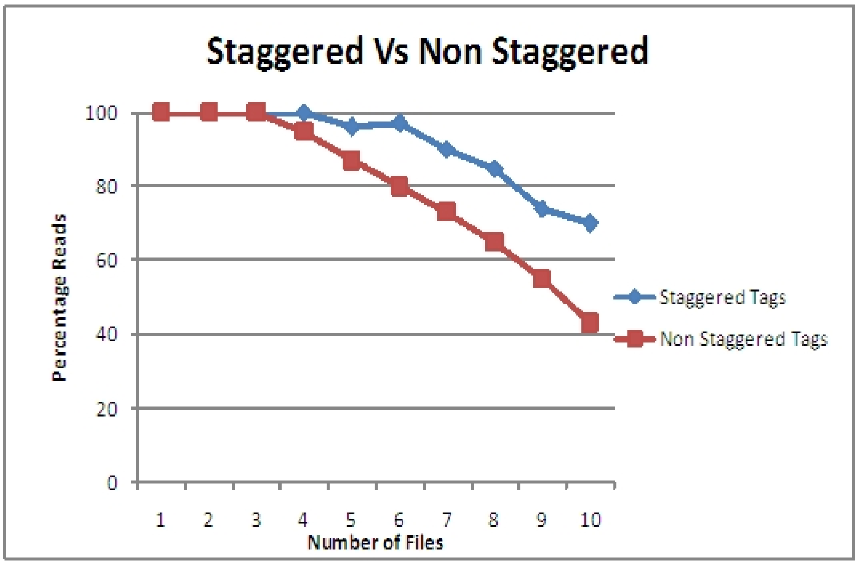

Figure 4.7: Comparison between Staggered and Non-Staggered Positioning

An article to be sent, is first collected at the local post office. The local post offices act as the customer-end interface for small regions in the town. The articles collected at these local post offices are then carried over to a mail sorting office of the town. In the mail sorting office, all articles for a specific destination are put into a single bag. A packing list is maintained for each bag. A paper tag specifying the destination is attached to this bag. Hereon these paper tags are used to identify the bags. The bags from the mail sorting office are then carried over to the mail transit office where they are dispatched to their respective destinations. The reverse process is carried out at the destination mail transit office. In this existing system, the article is logged only at the source mail sorting office and after the final delivery to the end customer. The visibility of the article is very poor during its transit from the source to the destination. In cases, where the bag moves through various transit offices, it becomes impossible to track the exact location of the bag and thus the article. This incompetence may lead to chaos, if a bag is lost during the transit.

4.2.2 Proposed RFID system

We have developed an RFID based system to track the movement of these bags. Every bag which is created at the mail sorting office is RFID tagged with the following information – bag-ID, source town, destination town, date and time of creation of the bag. Information about the articles (article-ID) contained in the bag is stored at a central location at the time of packing list creation. We decided to introduce the RFID tracking points at the mail sorting offices and mail transit points. These are the places, where all bags are aggregated/segregated according to the destination. The RFID readers are placed at these tracking points. At the mail sorting offices, the reader antennae are placed at the entry and exit. This helps to track the incoming and outgoing bags. Mobile RFID readers are used at the mail transit offices for the reasons of flexibility. These mobile readers read the bags coming in or going out of the mail transit office. The applications at each of these locations are connected to the central database to which all the tag reads are updated periodically. The end user is provided with a web based interface to query the article. The query uses the article-bag mapping and the RFID log at central database and displays the route history of the article.

The implementation of this system is successfully carried out in our lab (figure 4.8and figure 4.9).

4.2.3 Implementation Details

A web based application is developed for tracking the bags and articles. During the creation of the tag, the bag details (bag-ID, source town, destination town, date and time of creation) is stored in the database. All data read from the tags are written in the central database. On a query for a particular article, the application displays the list of events for that article. These events include the entry and exit of the article at different location on its route. The components involved in the development of this system are the following.

RFID Hardware – EPC Gen2 Tags (96 bit UID), UHF RFID Readers.

Software – SmartRF – The RFID middleware, Generic Application Framework, Central Database implemented using MySQL Community Server, Apache HTTP Web Server.

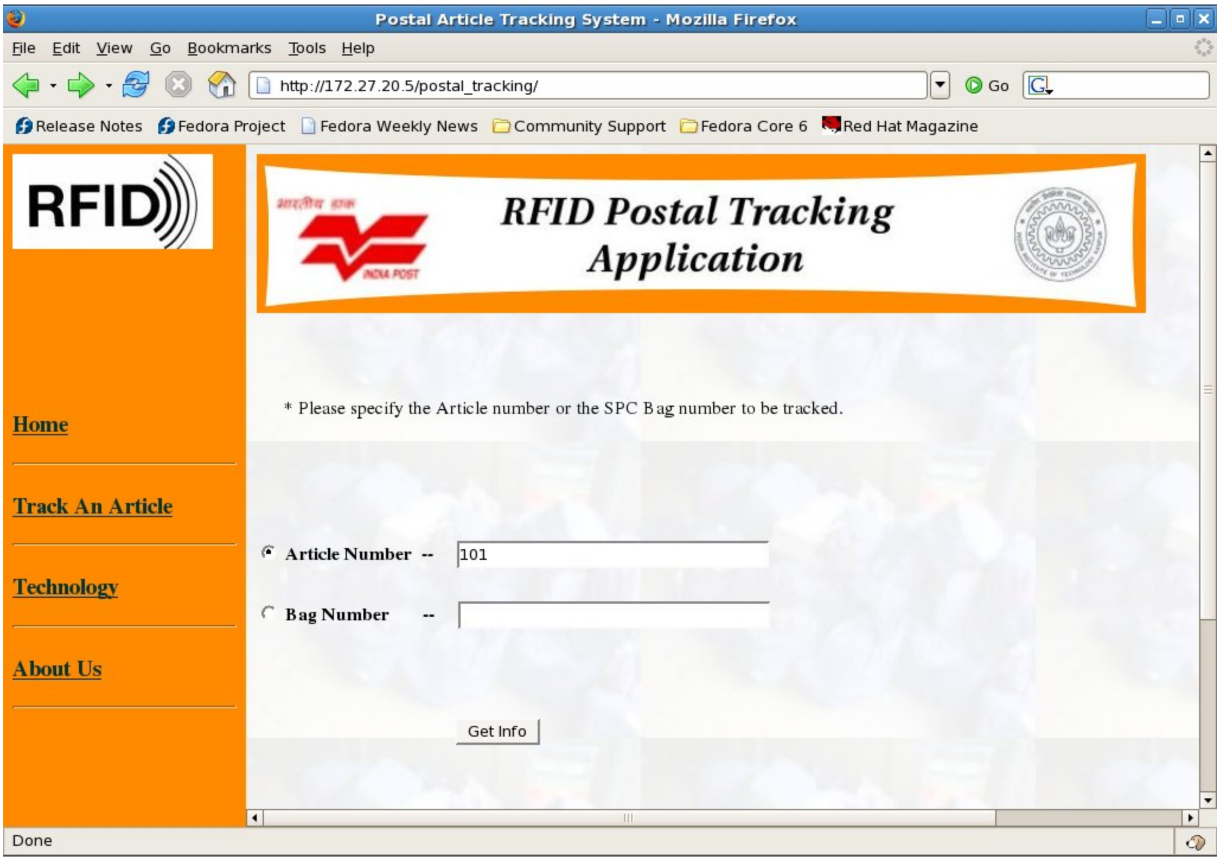

Figure 4.8: Portal for Postal Article Tracking (I)

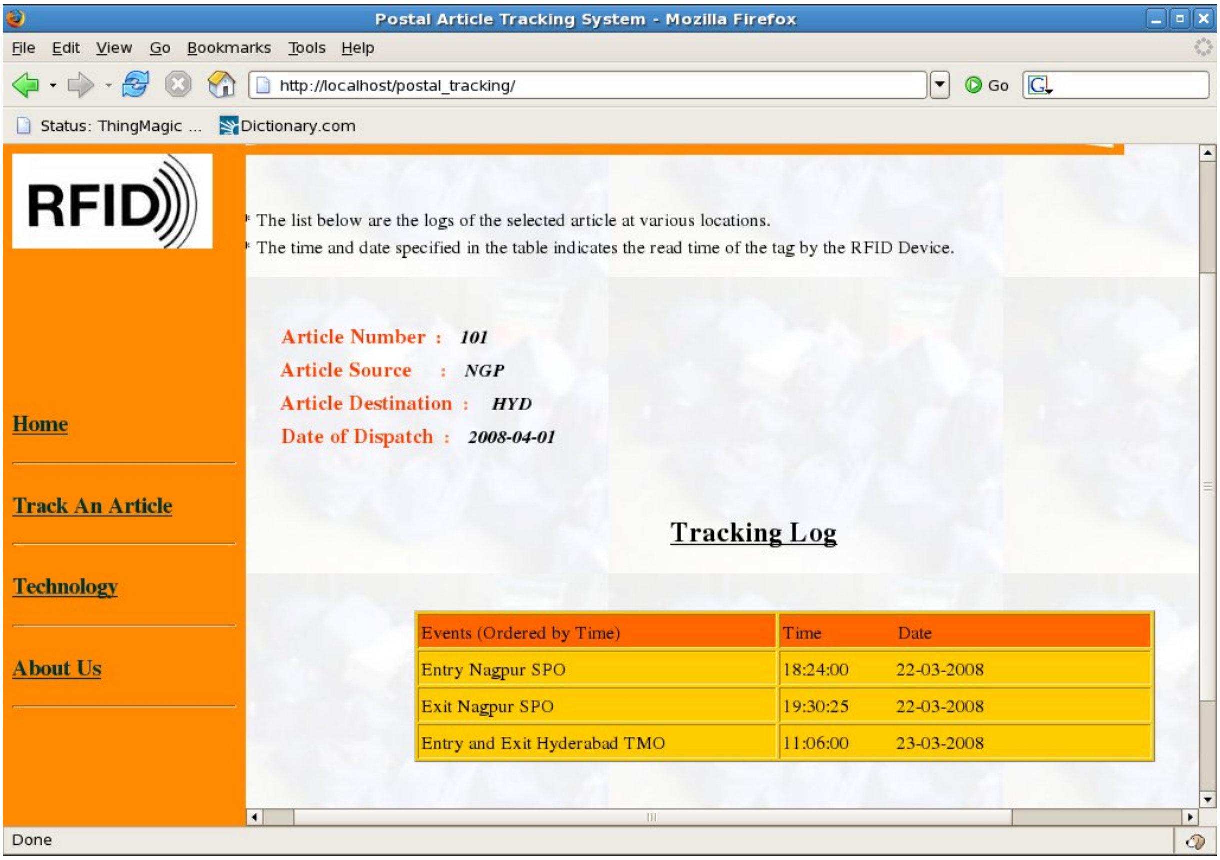

Figure 4.9: Portal for Postal Article Tracking (II)

Chapter 5

Conclusion

The RFID based systems are excellent options for deployment in enterprise applications for automatic tracking and identification purposes. However, the development of these systems is not a simple process and requires enormous processing of data from the RFID hardware devices in addition to the implementation of business logic. There are lots of hardware related parameters to be configured and there are hardware related intricacies involved in interfacing with the RFID hardware. A software layer is, therefore, needed between the enterprise applications and the RFID hardware. This layer, known as the RFID middleware, makes the interfacing of applications with the RFID devices easier. Even with this layer there is good amount of processing related to the sequencing of events to implement business logic. The application framework developed as part of this work accomplishes these requirements. The framework provides the data to applications in form of a sequence of bytes. The advantage of providing data in byte sequence permits the framework to be generic. It can be used with any kind of application and with any data exchange protocol (EPCIS, etc).

The applications developed by us with our application framework gave an insight into the physical aspects in deploying the RFID based systems. The most disturbing factors are the inability of the readers to detect tags reliably everywhere and any time. The effect of human presence and the presence of metallic objects in the vicinity are other aspects. Most of the time the tags were not detected when they were stacked one over the other. This restricts the number of bags and files which can be passed through RFID gates in one go. The human beings also absorb RF energy limiting the illumination of tags held close to the body. This puts additional restrictions on the way the files/bags can be moved across the RFID gates.

We carried out experiments on staggering the tags on the physical placement to avoid stacking of these tags even when the files are stacked one on top of other. This resulted in higher reliability and better success rates.

The usage of regular expression to identify the occurrence of events helps in easier specifications and implementation of various complex business rules. It also resulted in error recovery in case some tags were not read.

The application framework can be used to develop many different types of applications with varying requirements in several different business environments. These applications can range from simple access management to complex warehouse management. The ease of use of this application framework helps easy development and deployment of applications.

Bibliography

[1] www.nxp.com/products/identification/.

[2] American National Standards Institute. http://www.ansi.org/.

[3] Bar Codes. http://en.wikipedia.org/wiki/Bar_codes.

[4] Bureau of Indian Standards. http://www.bis.org.in/.

[5] Deutsches Institut für Normung. http://www.din.de/.

[6] EPC Global Inc. http://www. epcglobalinc. org/.

[7] EZ-Pass Inter Agency Group. http://www.e-zpassiag.com/.

[8] Pharmaceutical Product Tampering News Media Factsheet. *HDMA), year=April 2004*.

[9] Registered jack . http://en.wikipedia.org/wiki/Registered_jack/.

[10] Single Sign-On. http://en.wikipedia.org/wiki/Single sign-on.

[11] Gas up, Without Opening Your Wallet. *Business Week*, March, 1997.

[12] ISO 11784:1996. Radio frequency identification of animals – code structure. 1996.

[13] ISO 11785:1996. Radio frequency identification of animals – Technical concept. 1996.

[14] ISO 14443-3. Identification Cards - Contactless Integrated Circuit Cards - Proximity Cards, Part 3: Initialization and Anticollision. February 2001.

[15] ISO 15693-3. Identification Cards - Contactless Integrated Circuit Cards - Vicinity Cards, Part 3: Anticollision and Transmission Protocol. April 2001.

[16] ISO 18000-6. Information technology - Radio frequency identification for item management. August 2004.

[17] R. Angeles. Rfid Technologies: Supply-Chain Applications and Implementation Issues. *Information Systems Management*, volume 22:51–65, 2005.

[18] Z. Asif and M. Mandviwalla. Integrating the Supply Chain with RFID: A Technical and Business Analysis. *Communications of the Association for Information Systems*, Volume 15:393–427, 2005.

[19] Electronics Industries Association. EIA Standard RS-232-C Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Data Interchange . August 1969.

[20] R.W. Boss and American Library Association. *RFID Technology for Libraries*. American Library Association, 2003.

[21] J.H. Bowers and T.J. Clare. Inventory system using articles with RFID tags, 1999.

[22] A. Brewer, N. Sloan, and T.L. Landers. Intelligent Tracking in Manufacturing. *Journal of Intelligent Manufacturing*, 10:245–250, March 1999.

[23] M. Chung, J. Choi, K. Lee, and S.K. Rhyoo. Constructing Enterprise Application Framework for Secure RFID Application Using SPKI/SDSI. *Sixth International Conference on Advanced Language Processing and Web Information Technology, 2007. ALPIT 2007.*, pages 572–577, August 2007.

[24] Federal Trade Commision. Radio Frequency Identification: Applications and Implications for Consumers. Technical report, Federal Trade Commision, March 2005.

[25] B. Eckfeldt. What does RFID do for the consumer? *Communications of the ACM*, 48:77–79, September 2005.

[26] EPCglobal. EPC Information Services (EPCIS) Version 1.0 Specification. *www.epcglobalinc.org/standards/*, 2004.

[27] EPCglobal. Object Naming Service (ONS) Version 1.0. October 2005.

[28] W.J. Eradus and M.B. Jansen. Animal identification and monitoring. *Computers and Electronics in Agriculture*, pages 91–98, November 1999.

[29] EPC Class 1 Gen2. EPCglobal Tag Data Standards Version 1.3. March 2006.

[30] B. Glover and H. Bhatt. *RFID Essentials.* O'Reilly, 2006.

[31] E.P.C. Inc. The EPCglobal Network: Overview of Design, Benefits and Security. *EPC Global White Paper*, 2004.

[32] M. Karkkainen. Increasing Efficiency in the Supply Chain for Short Shelf Life Goods using RFID Tagging. *International Journal of Retail & Distribution Management*, pages 529–536, October 2003.

[33] R. Koh, E.W. Schuster, I. Chackrabarti, and A. Bellman. Securing the Pharmaceutical Supply Chain. *Auto-ID Center MIT, White Paper, September*, 2003.

[34] J. Kohl and C. Neuman. RFC 1510: The Kerberos Network Authentication Service (V5), September 1993.

[35] W.C. Maloney. Object tracking system with non-contact object detection and identification, July 31 2007.

[36] M. Pappu, R. Singhal, and B. Zoghi. RFID in hospitals: issues and solutions. *white paper of Consortium for the Accelerated Deployment of RFID in Distribution,*, September, 2004.

[37] Speed Post. Department of Post, Govt Of India. March 2008. http://www.indiapost.gov.in/ Netscape/SpeedPost.html.

[38] M.R. Rieback, G.N. Gaydadjiev, B. Crispo, R.F.H. Hofman, and A.S. Tanenbaum. A Platform for RFID Security and Privacy Administration. *Proceedings of the 20th conference on Large Installation System Administration Conference-Volume 20*, pages 89–102, December 2006.

[39] K. Traub, S. Bent, T. Osinski, SN Peretz, S. Rehling, S. Rosenthal, and B. Tracey. The Application Level Events (ALE) Specification, Version 1.0. *EPCglobal Proposed Standard*, February 2005.

[40] H. Vogt. Multiple object identification with passive RFID tags. *IEEE International Conference on Systems, Man and Cybernetics*, October 2002.