16

RFID Data Management

Sapna Tyagi[1], M Ayoub Khan[2] and A Q Ansari[3]
[1]Institute of Management Studies, Ghaziabad
[2]Centre for Development of Advanced Computing, NOIDA
(Ministry of Communications and IT, Govt. of India)
[3]Department of Electrical Engineering, Jamia Millia Islamia, New Delhi,
India

1. Introduction

An Auto-Identification (Auto-ID) technology has achieved significant growth in various verticals as industries, purchasing and distribution technologies, manufacturing companies, defence and material flow system. Auto-ID is the term used to describe the process of automatic data collection and identification that occurs in real-time. Automatic identification procedures exist to provide information about people, animals, goods and products in transit.

Radio frequency identification (RFID) is a general term that is used to describe a system that transmits the identity (in the form of a unique identification number) of an object wirelessly, using radio waves without any line-of-sight between tagged objects and readers (Auto-ID Technical report, 2002; Khan M Ayoub et al., 2009). RFID is based on Radio Frequency, which operates in different frequency bands viz. low frequency (LF), high frequency (HF) and ultrahigh frequency (UHF). The frequency band is determined by the requirement of application like read range, sensitivity etc. The RFID Devices are capable of operating in frequency ranging from 125 KHz to beyond 2.5 GHz, however, due to regulatory restrictions (typically country specific) on use of radio–frequency spectrum, only few frequencies are commonly used. The two most common are 13.56 MHz in the HF band and frequency around 900 MHz in UHF band. The frequency affects the characteristics of the resulting sensing environment, for instance HF signal propagate more easily through plastic, paper and moisture as compared to UHF, where as UHF signals having longer range. RFID tags may derive the energy to operate either from an on-tag battery or by scavenging power from the electromagnetic radiation emitted by readers. The most obvious technology that is comparable to RFID for many application areas is Barcode. Both these technologies involve the addition of a 'tag' or 'label' to an item that contains information about that item which allows it to be identified by a computer system.

A system designed to identify objects based on RFID tags has two advantages over conventional Barcode systems:

1. Barcodes are fixed once they have been created, whereas the data contained within an RFID tag can typically be augmented or changed as required. This means that:

Source: Radio Frequency Identification Fundamentals and Applications, Bringing Research to Practice, Book edited by: Cristina Turcu, ISBN 978-953-7619-73-2, pp. 278, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

   a. It is possible to separate the time at which an object is tagged from the time at which information is stored on the tag – it may be advantageous, for example, to apply the tag at some point in an item's manufacturing process, before the information to be associated with the tag is known. This is not possible with a Barcode.
   b. Information can be updated as a tagged item moves through a process, keeping the important information with the tag (and the item) and so making it available at any point in its lifetime.
2. Barcodes have to be scanned deliberately by a person in a process that is difficult to automate. RFID tags, on the other hand, can be readily scanned automatically without human involvement. This reveals that:
   a. The data can be obtained continuously and thus they are more up-to-date than data obtained only at specific intervals (like inventory counts) and specific points in the supply chain (like shipping or receiving)
   b. Not involving a human in the process means that the readings can be less expensive and generally more accurate – incremental readings are virtually cost-free once the system has been set. It also means that there may be fewer misreads.
   c. Speed – many tags can be read simultaneously into a computer, rather than reading a single tag at a time.
3. Barcodes require line-of-sight to read, while RFID tags can be read (in any orientation) as long as they are within the reader's range. This implies that:
   a. The content of various conveyances (such as trailers, cases, pallets, shopping carts) can be read automatically without opening and sorting the conveyance.
   b. Barcodes do not work well when exposed to weather elements, when dirty, or if damaged in any way that interference with clear line-of-sight reading. RFID is much more suited to operation in harsh environments.

This chapter explores fundamentals of data management in RFID applications so that the data retrieved out of RFID application is non-redundant and filtered. The chapter is organized as follows: characteristics of the RFID data is discussed in section 2. Section 3 presents data flow and modeling of RFID data. Section 4 explores the scope of data warehouse for RFID data. This section also focuses on the filtering of noisy data. Section 5 has a focus on the role of RFID middleware. Finally, a conclusion and further research directions is presented in the last section.

2. RFID data characteristics

The RFID systems are being used in variety of applications; despite of this diversity, the data generated out of the RFID application share some common characteristics. The characteristics of RFID data is as follows (Oleksandr Mylyy, 2006):

Uniformity in data: Data emerging out of an RFID application is in the form of (tag_id, reader_id, timestamp) where tag_id and reader_id are the identification code that uniquely identifies the object and reader.

Tremendous amount of Data: This is one the biggest concern as RFID system generates terabytes of data in single day on an average. Thus to deal with such flood of data is challenging task. Adopters of RFID technology must ensure that their IT systems are dimensioned accordingly.

Time based: Each RFID observation is associated with timestamp. In such scenario, it is prerequisite to deal with order and sequence of the RFID observation in which they arrive.
Noisy Data: The RFID system has parallel transponders and receivers. Therefore, the huge information is generated every second that may not be reliable. Sometimes, the information generated from RFID tag is mingled with other tag values or other environmental hindrances.
The challenging issue in the RFID is to identify the presence of tag correctly because RFID environment is not clean. Therefore, let us classify the type of observations that the reader encounters which are as follows (Selwyn Piramuthu et al, 2008).

1. True Positive Readings: These readings refer to the case where the reader identifies the tag to be present while it is in the field of reader.
2. True Negative Readings: These readings refer to the case where the reader is reading the tag as being absent and it is truly not in the field of the reader.
3. False Negative Readings: In this Scenario, the reader might not detect RFID tags, which are in the vicinity of readers. This situation can be raised because of collision of RF signals or due to physical hindrance in the environment.
4. False Positive Readings: In an RFID system there are parallel transponders which have to be detected, so many times while reading one tag, unexpected extra readings from other tags is mixed which leads to inaccurate data.
5. Duplicate Readings: The same tag generates duplicate readings due to multiple readings cycle and multiple readers.

Readings	**Tag presence**	**Acceptance**	**Narration**
True Positive	+	True	Tag is being identified
True Negative	-	True	Tag is not being identified as tag is not in field of reader
False Positive	+	False	Tag is being identified but reported as inaccurate data
False Negative	-	False	Tag is not being identified even it is in field of reader
Duplicate readings	+	True	Read by multiple readers

Table 1. Observation (Reading) types

2.1 Electronic Product Code (EPC) format

RFID tags used in the supply chain are encoded with an Electronic Product Code, or EPC, which is a globally unique identifier for the object being tagged. There are a number of different encoding formats; which one a particular tag uses depends on the tagged item. These formats can be specific to groups of goods, such as shipping containers, or can be specific to each individual asset type. To ensure that each EPC is unique, EPCglobal (the organization driving standards for EPC) assigns each company a unique manager number.

Each company is then responsible for assigning the other fields required by the encoding format being used (EPCglobal, 2005).
EPC, the unique code across the globe, is stored into RFID tag's memory. The code is generic and follow universal numbering scheme for physical objects. The EPC identifies every single, individual product item where barcode only identifies the product. A 96 bit EPC code has structure as follows (EPCglobal, 2005; Khan, M Ayoub, 2009):

Header	EPC Manager	Object class	Serial number
8 Bit	28	24 bit	36 bit

Table 2. EPC Structure (96-Bit)

The first field in the header defines the coding schemes in operation with the remaining bits providing the actual product code. The Manager Field is responsible for identifying the product manufacturer. The object class defines the product class itself. The Serial number is unique for an individual product class. The length of EPC may be of 64, 96, 128,256,1K, 4K Bits. The 96-bit EPC can identify 268 million manufacturers uniquely. Each manufacturer can have 16 million unique object classes and 68 billion unique serial numbers within each object class. Such tags typically operate on the UHF band and are popular with retail and distribution industry (e.g. Wal-Mart). These 96-bit tags are commonly used because of their low cost. Other application may demand HF tags with enhanced capabilities. For example the airline industry is using HF tags, that can easily operate in the environmental extremes. These tags store not only an EPC but also the supplementary data such as repair and service history of part. There are various types of EPC class of different bit-length as shown in table 3.

Class	EPC Length (Bits)
Class-0	64
Class-I	96
Class-II	128/256
Class-III	256-1Kb
Class-IV	4K

Table 3. EPC Class (Khan, M Ayoub, 2009)

RFID readers typically return the raw HEX or binary representation of an EPC, values which must be decoded using bit-level programming to derive a useful representation of the information that a tag holds. As an example, an RFID reader may read and output a HEX value of 30700048440663802E185523. This value must be converted to binary, then decoded programmatically according to the EPC specification to extract the decimal field values, and finally, formatted to return a meaningful representation of the EPC called the Uniform Resource Identifier (URI) representation. The binary representation of the tag HEX value is shown as follows.

0011000001110000000000000001001000010001000000011001100100000000000000010111000011000010101010100100011

The EPC tag specification outlines the decoding process, which you can follow by interpreting the binary string bit by bit to get a more useful representation. After decoding the binary value, the URI representation of the tag above is as follows:

urn:epc:tag:sgtin-96:3.0037000.06542.773346595

That value must be processed further to determine the item it actually represents. The URI representation is often used for reporting as it is easier for programs or individuals to extract meaningful information about the tagged item from that representation than from HEX or binary values, by filtering or grouping on the various fields.

3. RFID data flow and modelling

The flow of RFID data in the supply chain management is an important aspect of modelling. The RFID tagged object moves from one location to another intermediate locations. At every location, RFID tag identity is matched with the related business data (i.e. requisition document number) in the receiving system so when a RFID tag identity is read, it can be processed further as an automated business event.

To describe this data process further, one needs to understand that the data flow process is composed of two subsets. The first subset is the transmission of RFID data from source to intermediate facility centres. The second subset is the transmission of RFID data from intermediate facility centre to destination.

3.1 Data flow in RFID

In an RFID system, there are two basic categories of data: inactive data and active data. Inactive data are related to commercial entities and product/service groups, such as location information, product level and serial level information. Active data are specific to individual items. There are two types of active data: instance data such as serial number and date of manufacture, and temporal data such as observations, location and containment changes of objects, which are all captured through EPC-tag readings. Among all the data, the temporal data are directly related to the fundamental business logic in RFID applications such as the movement and transaction of products. By examining RFID system, we summarize the following primary entities that interact with each other and generate business processes.

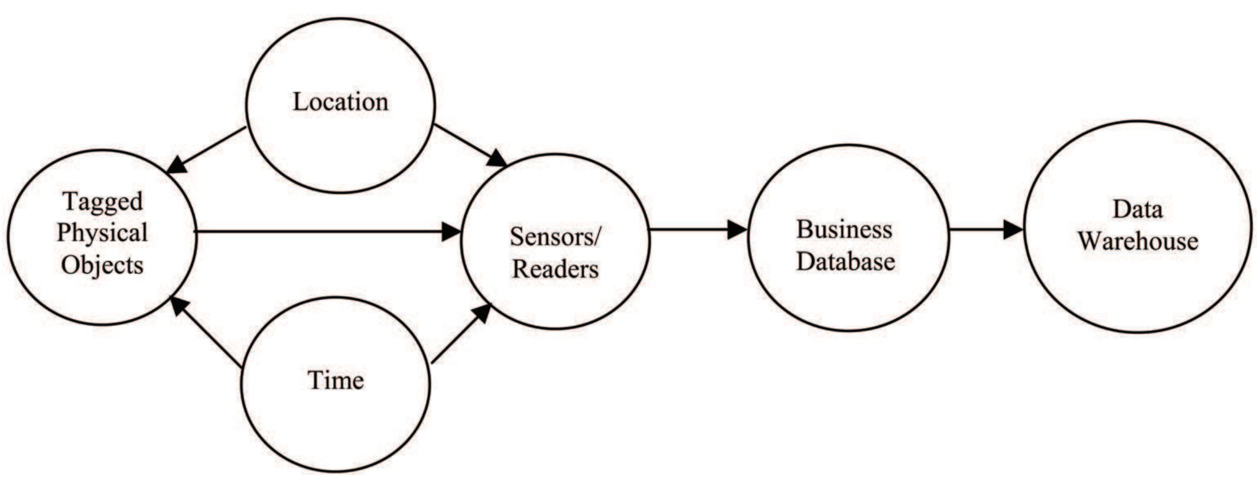


Fig. 1. Data flow in RFID

Objects: These include all EPC tagged objects as items, pallets, cases, and trucks.

Sensors/Readers: RFID readers use radio frequency signals to communicate with EPC tagged objects and read the EPC values. The reader also has unique identification code.

Locations: A location determines the current position of EPC tagged objects. It can be weaving factory, then distribution center and finally retail as shown in figure 2. The granularity of locations can be defined according to application needs.
Transactions: There can be business transactions in which EPC is involved. For example Checkout involves credit card transaction with many EPC readings.

3.2 RFID data modelling

Data modelling is a way to structure and organize data so that it can be used easily by the databases. Unstructured data can be found in word processing documents, email messages, audio or video files, and design programs. Data modelling doesn't want these "ugly" data; rather, data modelling wants data that is all made up in a nice, neat package for processing by a database. So in a way, data modelling is concerned with how the data looks.
Data modelling is routinely used in conjunction with a database management system. Data that has been modelled and made ready for this system can be identified in various ways, such as according to what they represent or how they relate to other data. The idea is to make data as presentable as possible, so analysis and integration can be done with as little effort as necessary. We can also think of data modelling as instructions for building a database. To make a "pretty" database, you have to follow a model as a means toward your desired end. Most of the defined entities (Objects, sensors, and readers) are static by nature but when they interact with each other they generate new observations that are required to be modelled along with static entities. These interactions generate an event that changes state. However, current RFID systems incorporate only event changes and state information has to predict implicitly. One of the essential goals of a RFID–enabled application is to track objects and monitor the system at any locations, at any time, or both. Thus, RFID applications require such data models that are enough capable of modelling these state changes and event changes. But, before explaining the RFID data model let us understand what are the general event changes and state changes (Fusheng wang & Peiya Liu, 2005).

3.2.1 State change

The value associated with the entities defines the state of the object. The change in the values leads to the change of the state as follows (Fusheng wang & Peiya Liu, 2005):
Object location change: For instance, the carrier (truck) and its loaded pallets leave the warehouse. This would change the location of the object.
Object Containment relationship change: Initially all the product items are packed into pallets, then all the pallets are loaded into truck as shown in figure 2.
Reader Location change: Reader 1 is installed at weaving factory, reader 2 is installed at distribution center and reader 3 is at retail.
Ownership change: Ownership changes as the product moves from manufacturer to retail.
Ownership location change: As location is always associated with owner so the detail is required to be captured. Thus, the information about during which period an object is in certain state is essential and has to be acquired.

3.2.2 Event transition

The events are generated time-to-time by the system or based on the interaction between reader and tag to accomplish a particular task as follows (Fusheng wang et al, 2005).

Observation: These are generated when readers interact with tagged objects.
Transacted items: These are generated when an object participates into transactions.

3.3 Dynamic temporal entity relationship model

Dynamic Temporal Entity Relationship Model (DTER) model is a dynamic and temporal based extension to ER model that can be efficiently used to model the entities and relationships discussed above with various new functionalities. There are two types of temporal relationships among RFID entities that we have already discussed: relationship that generates events and relationship that generate state history. For an Event based relation we use an attribute timestamp to represent the occurrence timestamp of the event. For a state–based temporal relationship, we use an attribute tstart and tend to represent the lifespan of the state. It also incorporates nested relationships like the application that uses Read/write tags, an onboard reader records the current temperature measurements or any other location based parameter like humidity etc in the tag. Thus, a reader observation contains both the EPC of the tag and the measurement history, i.e. nested relation

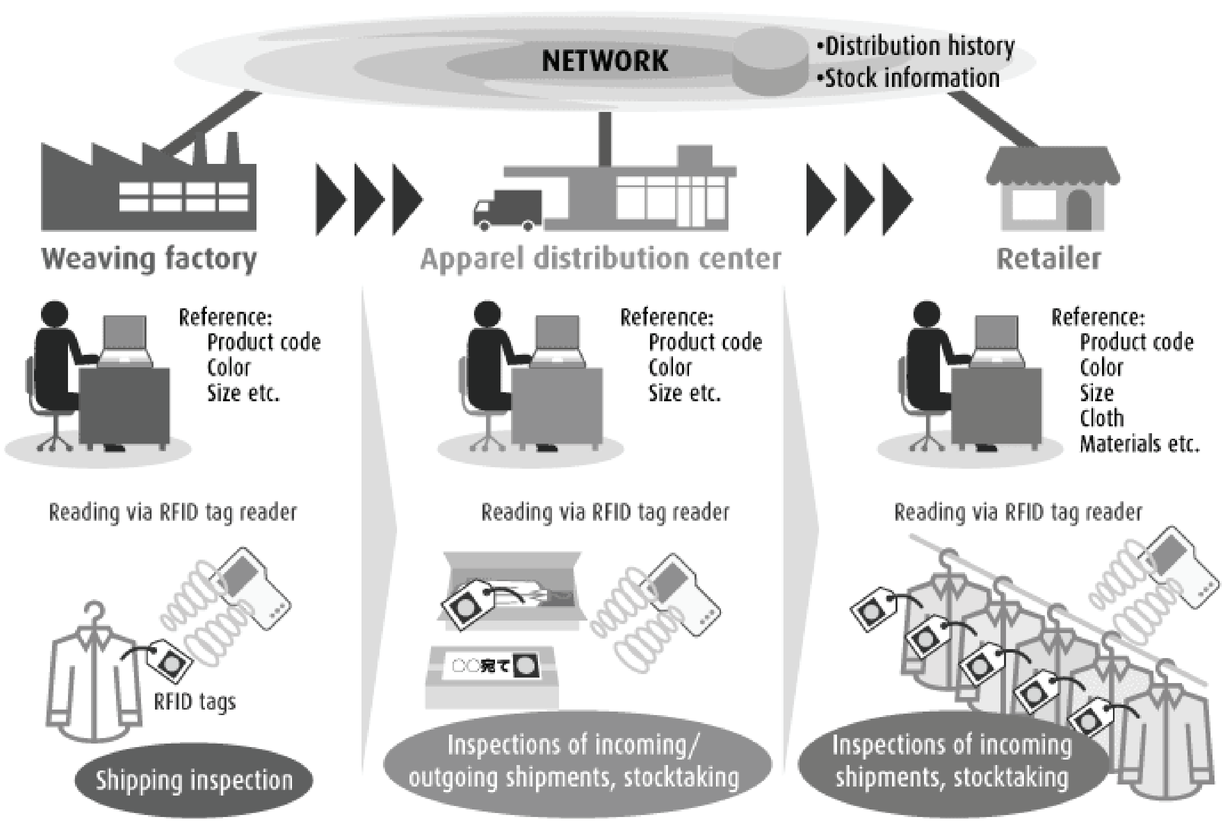


Fig. 2. Object movement chain

4. RFID data warehouse

Data warehousing in RFID is an emerging technology that facilitates in gathering, integrating heterogeneous data from distributed sources and extracting information that can be utilized as knowledgebase for decision support. The amount of information that would be generated by RFID tags is on the verge of exploding. RFID observation contains redundant, missed and unreliable data because of the various parallel transponders. Thus, it requires cleaning and filtering of the incoming data before warehousing. Thus, it requires cleaning and filtering of the incoming data before warehousing. Before discussing further about data warehousing, a brief architecture of RFID system, some filtering techniques and other data management practices must be well understood.

4.1 Architecture of RFID system

The Figure 3 represents layered system architecture of data movement in RFID environment. The lowest layer consists of RFID tags that are placed on the object to be identified such as cases or pallets.

The next layer is called Data Capture Layer (DCL). The data emerging from this layer can be considered as RFID data streams. They are usually in the form (tag id, reader id, timestamp). Both tag and the reader are identified using a global naming scheme called EPC (electronic Product code) by analogy with the UPC standard that is used for the bar codes.

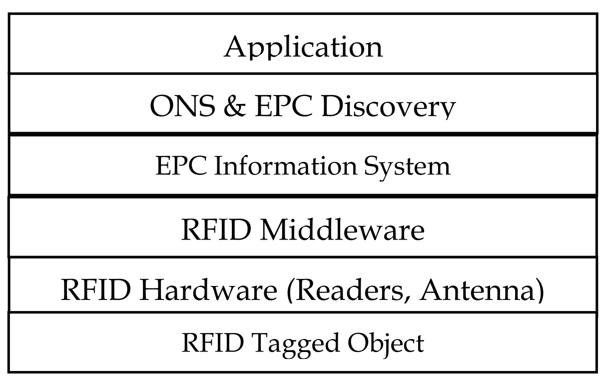


Fig. 3. Object movement chain

The third layer of the architecture is responsible for mapping the low-level data stream from readers to a more manageable form that is suitable for application level interaction. This is also called "savant" that is subject to standardization effort under the name middleware. This layer plays the primary role in RFID data management. RFID middleware systems typically deployed between the readers and the applications in order to correct captured readings and provide clean and meaningful data to application logic. In addition to cleaning data and coping with the idiosyncrasies of different kinds of reader. Application may interact with savants or middleware by issuing simple queries as well as by installing standing queries that result in a stream of matching data. We will study more about middleware in next section.

The fourth layer provides high-level service that is easier for application to use. For example this level maps EPC code to the type of object it represent (individual item, case, pallet) and provides information such as product names and manufacturers, It is also responsible for providing time specific information, such as expiration date of any frozen product represented by EPC code.

The fifth layer of the architecture is part of object Name Service (ONS). The ONS is essentially a global look up service that maps an EPC to a URL that describes the item represented by the EPC. The design of the ONS services uses NAPTR facility of the standard Domain Service (DNS) to rewrite EPC's into URLs. The mapping may be dynamic. For example, as a product moves from manufacturer to distributor and further down the supply chain, the ONS mapping changes to reflect the current custodian of the product.
The last layer of the architecture is the application layer where desired functionality achieved from the filtered RFID data. This application may be written in any high-level language using the library provided by the specific RFID reader vendor.

4.2 Filtering and cleaning RFID data

Due to the low-power and low cost constraints of RFID Tags, reliability of RFID readings is of concern in many circumstances. We have already discussed the type of reading that the reader encounters that leads to various undesirable scenarios i.e. false negative readings, false positive readings and duplicate readings.
In practice, readings are often performed in multiple cycles to achieve higher recognition rate. In this way, false negative readings can be significantly reduced. The noisy readings (or false readings) generally have a low occurrence rate compared to normal true readings. Thus, only those readings that have significant repeats within certain interval are considered to be true readings. However, this will produce much more duplicate readings. To understand the basics of multiple read cycles, a sliding window filtering technique is presented below.

4.2.1 Sliding window filtering technique

A sliding window is a window with certain size that moves with time. Suppose the window size has time coordinate of [t1, t1+window_size], after s time, the coordinate will become [t1+s, t1+window_size+s]. RFID reading tuples will enter the window and get expired as time moves. Therefore, the noise readings are reading with count of distinct tag EPC values below a noise threshold. Denoising essentially performs the following operations: within any time window with size of window_size surrounding an RFID reading, if the count of the readings with same tag EPC values appears equal to or above threshold, then the observed EPC is not noise and needs to be forwarded for further processing: otherwise the reading is discarded. Two parameter used here are window size of a sliding time window, and a threshold for noise detection.
An RFID observation (reading) is in the form of (reader_id, tag_id, timestamp) which refers to the EPC of the RFID reader, EPC of the tagged objects and the timestamp of the observation.

4.2.2 Baseline filtering technique

In this algorithm, intuitively, for each incoming reading of value R, we perform a full scan of the preceding sliding time window of size window size. If R appears more than threshold value within the window, then this is not a noise reading thus we output every R in the window. To ensure a particular reading is never output more than once, we keep a state-of output with each reading in the window buffer and set it to true once it produces the output.

```
Function Baseline denoise (window size, threshold)
        WINDOWBUFFER   empty queue;
        Loop {loop forever for next incoming reading}
        INCOMING   the next reading
        append INCOMING to the end of WINDOW- BUFFER
        EXPIRETIME   INCOMING.timestamp – window size
        while the head of WINDOWBUFFER is older than EXPIRETIME
        do
                remove the head of WINDOWBUFFER
        end while
                COUNT count of readings in WINDOW-BUFFER whose key equals to
                INCOMING.key
        if COUNT > threshold then
        for each of the reading R in WINDOWBUFFER
                with key equals to INCOMING.key do
                if R has not been output before then
                        output R
                        set STATE-OF-OUTPUT as true
                end if
        end for
        end if
        end loop
        print reading of value R, w
End Function
```

4.2.3 Dynamic threshold based Sliding-Window filtering

In order to reduce the false negative and false positive reading, all the existing literatures has discussed about the increase or decrease the size of window with some probability based on the circumstances. Some of them has considered concept of multiple readers. We approached this problem in a different manner. Consider a situation where threshold value t_h is six. This means after six occurrence of the raw data in a given time window period this will recognize it as a tag otherwise discard it by considering it as a noise. In the table 4 all black entries are the tag data and red entries are noise. If we consider, reading_of_tag_3 column, where the raw tag data "B2C1C2BA2FD1FA1E" occurs only three times while as the occurrence of noise is much larger and passes the threshold value. This increases false-negative rate. We are proposing following modification to sliding-window filtering (Y. Bai et al., 2006) technique.

1. Threshold value shall be updated periodically.
2. RFID data format and associate values (Header information) shall be examined, after recognizing it as a tag.

Former will help in changing the threshold value t_h as the environment will change. Typically, if the error rate is going up then t_h has to be decreased. Later would help in eliminating a noise that is being recognized as a tag. This happens because of sufficient number of occurrence of noise and passes through the threshold value.

Let's formulate the problem of filtering with few assumptions as follows:

S: window Size (In time domain)

S = r x i Where { r is count (repeat count); i is interval between readrepeats }

```
Struct EPCPacket
{
EPC EPCData;
Time T;
Reader R;
}
th: threshold value
e: error rate
DThreshold_SW_Filter(S, T)
{
        EPC Window_Buffer[S]; // Buffer holds EPC data
        Time ArrivalTime [];
        EPCPacket currentEPC;
        Integer EPCCount;
While (TRUE)
{
EPCList = CreatEPCList (EPCPacket); // create a list to hold EPCPacket
data=GetEPCReading ()// get next reading from reader
EPCList.AddEPCPacket (CurrentEPC);
LifTimeofEPCPacket = S – CurrentEPC.T;
EPCCount= FindEPC (currentEPC.EPCData ());
If (EPCCount >= th)
{
At= GetArrivalTime ();
ArrivalTime [] =At; // this will preserve the out-//of-order // sequence problem
If (CheckEPCHeader (CurrentEPC) == TRUE)
        {
        Sort (ArrivalTime [])
        Print (ArrivalTime);

        }
Else
        {
        EPCList.RemoveEPCPacket(LastEPC)
        //Remove that data from the EPC list, because it is a noise
        }
}
If(GetErrorRate() > e)
{
Decrease(th);
}
}//end-of-while
}//end-of- DThreshold_SW_Filter function
```

In the algorithm, to preserve the arrival sequence we have introduced a queue to store the entry time of the tag with the help of GetArrivalTime( ). Then this queue is sorted. The problem of noise is eliminated by introducing CheckEPCHeader ( ) that will examined whether a tag is a tag in actual or a noise. If the value returned is true then only it will display as a tag otherwise discard it. Threshold value th as the environment will change. Typically, if the error rate is going up then threshold has to be decrease. The data for RFID has been generated with the help of EPC Generator. This is variable length EPC data generator i.e. 64, 96, 1128, 256, 512 etc. But, we have generated 64-bit data as show in table 4.

Time	Reading_of _Tag_1	Reading_of_Tag_2	Reading_of_Tag_3
0	FF3CD4FB8ED4FB8E		
100	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	
200	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	A1B4C2BA2FD1FA1E
300	3FFCE1FC5FA11C8E	CC4FC3AC2FD1FE8E	3FFCE1FC5FA11C8E3
400	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	3FFCE1FC5FA11C8E3
500	FF3CD4FB8ED4FB8E	3FFCE1FC5FA11C8E	3FFCE1FC5FA11C8E3
600	3FFCE1FC5FA11C8E	3FFCE1FC5FA11C8E	3FFCE1FC5FA11C8E3
700	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	3FFCE1FC5FA11C8E3
800	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	3FFCE1FC5FA11C8E3
900	FF3CD4FB8ED4FB8E	FFFCE1FF5FA11B8E	3FFCE1FC5FA11C8E3
1000	FF3CD4FB8ED4FB8E	FFFCE1FF5FA11B8E	FFFCE1FF5FA11B8E3
1100	3FFCE1FC5FA11C8E	CC4FC3AC2FD1FE8E	A1B4C2BA2FD1FA1E
1200	3FFCE1FC5FA11C8E	CC4FC3AC2FD1FE8E	A1B4C2BA2FD1FA1E
1300	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	A1B4C2BA2FD1FA1E
1400	FF3CD4FB8ED4FB8E	FFFCE1FF5FA11B8E	A1B4C2BA2FD1FA1E
1500	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	A1B4C2BA2FD1FA1E
1600	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	A1B4C2BA2FD1FA1E
1700	FF3CD4FB8ED4FB8E	FFFCE1FF5FA11B8E	B2C1C2BA2FD1FA1E
1800	FF3CD4FB8ED4FB8E	CC4FC3AC2FD1FE8E	B2C1C2BA2FD1FA1E
1900	3FFCE1FC5FA11C8E	CC4FC3AC2FD1FE8E	B2C1C2BA2FD1FA1E

Table 4. RFID raw data generated

4.2.4 Multiple readers filtering technique

It's an approach different from the sliding window that caters problem of false reads. This technique is based on the belief that a certain extent of the false reads problem can be caused when communication between tag and the reader is achieved somehow regardless of the presence of signal-blocking entities such as metal shielding. For example a tag might be "visible" to a reader at one orientation but might be "invisible" to a reader in another orientation because the obstacle (e.g. metal shielding) affects communication between tag and reader in one orientation but not in the other. Hence the method deploys multiple readers or tags in order to take advantage of varied signal orientation. The basic idea is that tagged object can be identified to be confirmed as present or absent if consistent reads are generated by both readers: otherwise the tagged object is identified using a pre-determined probability P.

4.2.5 Duplicate elimination

When noise in the readings is eliminated, duplicate readings for the same tag have to be recognized and only the first one among all duplicates should be retained. The duplicate elimination algorithm takes parameter Max-distance in time domain. If a reading is within max distance in time from the previous reading with the same key (reader id, tag id), then this reading is considered a new reading and is output.

Raw RFID Records
(r1; l1; t1) (r2; l1; t1) (r3; l1; t1) (r4; l1; t1) (r5; l1; t1) (r6; l1; t1) (r7; l1; t1) : : : (r1; l1; t9) (r2; l1; t9) (r3; l1; t9) (r4; l1; t9) : : : (r1; l1; t10) (r2; l1; t10) (r3; l1; t10) (r4; l1; t10) (r7; l4; t10) : : : (r7; l4; t19) : : : (r1; l3; t21) (r2; l3; t21) (r4; l3; t21) (r5; l3; t21) : : : (r6; l6; t35) : : : (r2; l5; t40) (r3; l5; t40) (r6; l6; t40) : : :... (r2; l5; t60) (r3; l5; t60)

Table 5. Raw RFID records

In order to reduce the large amount of redundancy in the raw data, data cleaning should be performed. The output after data cleaning is a set of clean stay records of the form (EPC, location, time in, time out) where time in is the time when the object enters the location, and time out is the time when the object leaves the location. Data cleaning of stay records can be

EPC	Stay(EPC; location; time in; time out)
r1	*(r1; l1; t1; t10)(r1; l3; t20; t30)*
r2	*(r2; l1; t1; t10)(r2; l3; t20; t30)(r2; l5; t40; t60)*
r3	*(r3; l1; t1; t10)(r3; l3; t20; t30)(r3; l5; t40; t60)*
r4	*(r4; 1; t1; t10)*
r5	*(r5; l2; t1; t8)(r5; l3; t20; t30)(r5; l5; t40; t60)*
r6	*(r6; l2; t1; t8)(r6; l3; t20; t30)(r6; l6; t35; t50)*
r7	*(r7; l2; t1; t8)(r7; l4; t10; t20)*

Table 6. Cleaned RFID records

accomplished by sorting the raw data on EPC and time, and generating time in and time out for each location by merging consecutive records for the same object staying at the same location. Table 6 presents the RFID database of Table 5 after cleaning. It has been reduced from 188 records to just 17 records (Hector Gongalez et al, 2006).

4.3 RFID data management practices

The amount of information that will be generated by radio frequency identification (RFID) tags is enormous. That leaves us with questions like "What happens to data quality? What data should we capture, and how often should we capture it? What about 'white noise'?" While we can't address every issue regarding the incoming data avalanche, we can highlight some of the more "front of mind" concerns surrounding RFID. In the effort to address these many issues, adopters of RFID technology are overlooking various important aspects of RFID deployment like how back-end databases and business application can handle the massive amount of new data that RFID systems will produce. In the rush to implement

RFID, users are overlooking the implication to their IT system. Too much focus is placed at present on the price of tags and abilities of readers but not enough on the data, how it is going to be used. If IT infrastructures are not updated to handle the new load they will suffer and shaky infrastructure would collapse.

4.3.1 Turning raw data into information

When designing an RFID system, we should first understand and consider two key aspects of turning RFID data into useful information. First, we need a way to convert the raw incoming RFID data into a meaningful context for further processing and subsequent actions. Because today's marketplace provides an abundance of RFID tag choices, data encoding formats, and custom data options, we'll need a powerful and flexible encoding and decoding architecture to support applications now and into the future. Second, while it might be relatively easy to build an RFID data acquisition and analysis system for the number of tags your business uses today, you have to consider the future. The system must be able to avoid data overload when your system collects data from hundreds of thousands of RFID tags. Filtering and smoothing are important concepts to understand; early in the design process you need to identify architectures that provides flexibility in processing data at the point of activity (M. Palmer, 2004).

4.3.2 Well defined business processes

Let business requirements drive the collection of RFID data. It should be up to the business managers to define what constitutes a business event and how this translates into a read or write transaction on a tag. Adjust the frequency of read or write events to the needs of the business. For example, asset tracking within an Army maintenance and repair facility may require tracking items as they move from one maintenance and repair service area to another, as opposed to installing readers on shelves where parts are stored. In other words, the granularity of data collection should be driven by business requirements not by what is technically possible (M. Palmer, 2004).

Try not to deploy an RFID system directly connecting RFID readers to your central IT systems that may lead to disaster. A better approach is to digest your RFID event traffic close to the source i.e. at the edge of the enterprise and forward only meaningful events to central IT system. For that, devices like network routers and hubs will need to become “smart” and run filters to get rid the network of bad feeds and undesirable information. Transformation will occur in two stages: on the transponders (readers) themselves and on the warehouse receiving the RFID transmission information, Time and date stamping will move to the forefront of database processing necessity (M. Palmer, 2004).

4.3.3 Transform simple data into meaningful data

A simple data stream has to be converted into meaningful data streams that can be directly stored into database. This Process is known as Complex event processing in RFID environment that extract actionable knowledge from discreet events.

4.3.4 Determine business rule

While RFID technologies have the ability to provide a massive amount of data, the first step in a successful data management strategy is to ensure that only meaningful information is passed on from the edge server-the server connected to the readers-to your back-end applications and data repositories. It's critical that business rules are well defined up front to help separate meaningful information from unwanted data as close to the readers as possible. This will help to reduce the burden on the network and on data storage systems. Additionally, since RFID opens up the capability for unique item-level tracking as opposed to tracking at the part number or stock-keeping unit (SKU) level, changes may be necessary to both data repositories and to enterprise applications in order to accommodate this level of granularity. Determine changes to database schemas that will help ensure that events and attributes specific to unique items can be accurately captured. In addition to data about unique items, you will also want to prepare for data about your RFID infrastructure itself. Information about the physical location and settings of your RFID reader infrastructure must be carefully managed so that you can correlate events with physical locations.

4.3.5 Leverage existing architectures and framework for data integration

When integrating RFID data with enterprise applications, one of the challenges is to leverage the infrastructure already in place to minimize costs. This challenge can be addressed by using existing service-oriented architectures and vendor-neutral integration frameworks to help provide the appropriate IT and business services. Be sure to delineate business services and IT services within your technical architecture. The IT services, such as routing and transformation, should provide a framework for integrating your data irrespective of your business scenarios. Your business services, such as specific business rules, can provide the layer of business functionality that uses the underlying technical service layer.

4.3.6 Data buffering

An RFID data concentrator is software which actually responsible for buffering of the data. There are three primary elements in it: RFID middleware, event processing and an in-memory data cache. RFID middleware provides the interface for applications to receive RFID data from readers. Event Processing handles high-volume, high-performance flows of data by organizing raw data into pipelines (M. Palmer, 2004).

4.3.7 Business dynamics, data ownership and privacy

Over the next three to five years, RFID will force companies to redefine the rules of engagement for collaboration in terms of the how supply chain data is exchanged and protected. Today, many participants in the supply chain are able to benefit from process inefficiencies that are subsidized by their partners. For example, manufacturers don't penalize retailers for stocking too many items within a store. Many stores actually receive merchandise credit plus a service fee when they return expired items to the manufacturer. Electronic tagging can help to smooth out the balance of power in the supply chain by increasing visibility into operations for all companies involved. Some will benefit; others may not. Be certain to understand the business dynamics across all entities that handle your goods or assets, and determine data ownership or privacy issues that may arise.

4.3.8 Analytical information

Having all the data in the world will not help improve a company's profitability if the data is not interpreted correctly or if no one is able to act upon it. Determine business requirements, such as key performance indicators, that will help you understand and take action based on the data that's collected by RFID systems. In some cases, such as in supply chain performance management scenarios, you may need to be alerted in real time or near-real time when key events or exceptions have occurred or are about to occur. In other cases, the response time may be less critical, and you can employ more asynchronous or batch-oriented techniques in order to process and interpret your data. A business intelligence dashboard can help to monitor key metrics and key processes and allow you to drill down into individual events and transactions for further detail. If you want to automate some of this sense-and-respond activity, consider emerging technologies, such as the Semantic Web standards, which can help computers better interpret data and take actions themselves. In essence, the Semantic Web standards allow computers to better understand the "meaning" of data; this is vital for improved search accuracy and for improved machine-to-machine automation of complex tasks. RFID technologies can provide information that gives companies a greater visibility into their supply chain and a better understanding of their operations, but the decision as to how to respond to this information is ultimately a human one.

4.3.9 Data context

Generally RFID data is simple and straightforward, unless we incorporate complex information using sophisticated, expensive tags; all we get is an identification number of the item, a time and location. Determining in detail data from simple data requires context and context typically comes from “Reference” data in a variety of forms. For example, context can come from information in advanced shipping notice (ASN) as provided a manufacture plant can use the ASN used to confirm that tagged items sent by the manufacturers when actually received.

4.3.10 Age RFID data gracefully

On an Average RFID system requires 7 terabytes of disk capacity daily to accommodate the RFID data, so it is mandatory for us to reduce data continuously, so that we can get manageable working set. The RFID middleware can delete data that are superfluous (e.g. redundant reads of tagged items at different location in the supply chain). Thus Age RFID data to keep your working set of data manageable, enrich raw data with required context, and reduce the load on down-stream systems.

4.4 Data warehousing for RFID applications

Data warehouse is a subject-oriented, time-variant, nonvolatile and integrated technique for organizing data. The components of a data warehouse are the source systems, operational data stores (ODS), data warehouses, data marts, Business Intelligence (BI) reporting and analysis tools, and data movement tools (also called ETL tools). Both the RFID and BI/DW technologies get integrated seamlessly into an architecture that not only puts all the pieces of the jigsaw puzzle together but also helps to develop other tangible architectures in an enterprise framework. Shown below is one such architecture that depicts the use of multiple ODSs and a centralized DW for the capture, processing and analysis of the RFID data.

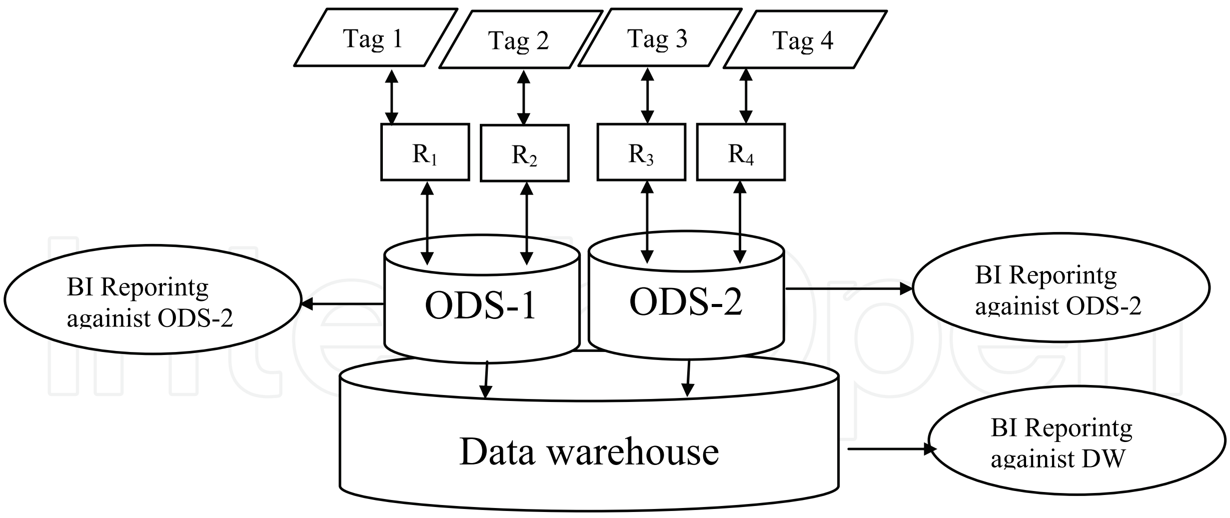


Fig. 4. RFID Data Warehouse and BI

In the above architecture (Figure 4) the data is captured via the RFID tags and passed onto the ODS via the RFID readers. The data from the distributed ODS is then moved to a centralized DW. BI reporting systems can be built to report against the localized ODS and also the central DW as per the reporting needs.
It is very interesting to note that both conceptually and technically there are a number of parallels between the RFID technology components and the BI/DW components and hence they neatly map onto each other, as shown in table.
Now that we have a proper architecture in place, let us discuss some of the BI/DW concepts that are going to be challenged by the onset of the RFID data into the DW/BI systems.
The most crucial point of impact of any RFID-enabled BI/DW initiative is the RFID data sources. RFID technology will bring in a dynamic exploding data in the data source layer of the DW/BI initiative. This will in turn demand a more dynamic analysis of the RFID data, for it to be of any use.

DW Dimensions and Facts	How RFID Answers it
Which (product)	Using the EPPC on the RFID Tag
When (time)	By the RFID Reader Device
Where (time)	By the RFID Reader device
Where (Geography)	By the RFID Reader Device)
Who (Customer)	By the RFID Reader/Tag combination
How much (Quantity)	By the RFID Reader/Tag combination

Table 7. RFID and BI/DW Parallels

The next crucial impact is on the data modeling, data organization and data interpretation technologies. There is a need for newer and efficient data modeling and DW architecting techniques to meet the influx of RFID data sources into DW/BI analysis system. This will also necessitate newer semantic-oriented database technologies and the appropriate data modeling techniques, to handle these large and constantly varying RFID data with a pertinent need for spatial (geographical) tracking of the RFID data across the supply chain of any business domain.

The third crucial impact is the data standardization of the RFID data, so that all the entities in the supply chain can automatically share data, thus enabling a tremendous increase in automation at many levels of the supply chain environment. It is here that the BI/DW technologies are far more matured and evolved compared to the OLTP systems and hence will play a crucial role in the data standardization processes at both syntax and semantic levels.
The fourth impact is the event-based approach to data and information analysis that the BI/DW will have to provide. This is because auto ID/RFID enables machine-to-machine communication, which is automatic, event-driven, and which necessitates that data is captured and processed in real time. This will have a big impact on the way BI/DW processes are designed, which means the BI/DW architectures will have to be flexible, agile, and efficient and event oriented.

4.4.1 Active data warehousing for RFID data

An active data warehouse presents an extension of the enterprise data warehousing capabilities. The foundation of ADW architecture is the detailed transaction history. Responding to near-time business events as they occur, completing complex analysis upon demand, and alerting people or systems to take action leverage the analytical capabilities offered by this infrastructure. What makes this different from the traditional use of data warehousing technology for business intelligence is that an active data warehouse is a closed loop system. Events are analyzed as they occur, and intelligent decisions are promptly initiated. Closed loop systems allow an organization to automatically respond to opportunities with agility, often without the need for human intervention. As business analytics become more instrumental in strategic decision-making, the Active Data Warehousing technology is maturing. This technology is engaged in integrating advanced decision support with day-to-day, even minute-to-minute decision-making that increases quality that encourages customer loyalty and thus secures an organization's bottom line. It may involve components like:

- A well-architect dimensional data model for the destination data mart
- A streamlined ETL process
- An aggressive sizing strategy
- A framework and architecture for expected reporting patterns

Active Data Warehousing extends the support from the traditional data warehouse by:

- Allowing access by customers, partners and suppliers at the same time
- Integrating multi-subject, cross-channel information
- Allowing fully detailed ad-hoc reporting and machine modelling

However, the task of accumulating the data from tags in RFID environment is similar as with analogous task in data warehouse, but still there are several differences in traditional data warehousing and RFID data warehousing. In traditional data warehouse, the data is not queried at the its source, The crux is on storing it into central warehouse, from there it will be indexed and queried, where as In RFID environment data generated by the tag reader is more likely to be used at the same place as of its origin which requires efficient transformation (cleaning, filtering, joining) before storing it into the warehouse.

4.5 Challenges in managing RFID Data

Large volume of RFID Data: RFID systems will have an unprecedented ability to produce great volumes of raw data in relatively short span of time. Adopters of RFID Technology must ensure their IT systems are dimensioned accordingly.

Requirement of Integration: The low level RFID observation need to be transformed and aggregated in semantically meaningful data suitable for corresponding application level. Enterprise with geographically distributed facilities networked to a central IT facility will be faced with the problem of managing raw RFID data while at the same time aggregating it into the central IT facility. Having large quantities of data flowing across network could place a burden on the enterprise IT infrastructure.
Data ownership and partner Data Integration: In retail supply chains or other applications in which data would need to be shared between different companies, questions might arise pertaining to the ownership of data. This could hinder integration of RFID systems between the companies.
Product Information Maintenance: In some application, retail supply chains for instances, central IT databases might continually need to be accessed to retrieve product information. In large-scale implementations, when high volumes of tags are processed, this could put extra burden on IT infrastructure.
Limited Communication Bandwidth: RFID system rely on the availability of unlicensed frequency bands, such as UHF is typically required in supply chain applications because of its increased read range, but European radio regulations permit the use of 865.0 MHz and 868.0 MHz by RFID readers. Another constraint is the bandwidth available per channel that limits data transmission rate between readers and tags. It restricts the number of tags that can typically be identified.

5. Role of RFID middleware

Radio Frequency Identification (RFID) holds the promise to automatically and inexpensively identifying items as they move through the supply chain. Tag and reader Physics solves the problem of being able to capture RFID Data. The Widespread adoption of RFID requires not only low cost tags and readers, but also the appropriate networking infrastructure. As already we have discussed the amount of information that would be generated by RFID tags is on the verge of exploding. RFID observation contains redundant, missed and unreliable data because of the various parallel transponders and to uncover various operational benefits of RFID, there must be something which can process the incoming data and intelligently integrate it into your application, as it is not a feasible solution to link existing software application directly to the RFID readers. The Reason Behind it is:

1. Inappropriate Incoming Data: Not all the incoming data is valuable. Duplicate reads and excess information must be filtered out so as not to bog down the network and end up with unstructured and confusing information within application.
2. Disparity between the Readers: Not all the readers speak same language, Building custom integration logic to each brand of reader will quickly eat up your RFID deployment teams time and budget.
3. Data Dissemination: Different RFID information needs to be passed off to different application and data stores.

Thus it requires cleaning and filtering of the incoming data before installing it into application. In this scenario RFID middleware plays a key role to achieve the maximum benefit of RFID technology. RFID middleware, simply put, is a software layer residing between the RFID hardware and the existing back-end system or application software. It extracts data from the RFID interrogators (readers), filters it, aggregates it and routes it to enterprise applications such as a warehouse management system (WMS), enterprise resource planning (ERP) software or a manufacturing execution system (MES). Many RFID middleware focused on various features like reader integration and coordination, EPC track and trace tools, baseline filtering capabilities, but these are just a subset of many features the complete RFID middleware platforms must provide.

5.1 Additional consideration for RFID middleware

Reader and Device Management: There are no standard, consistent tools to manage readers, check their health, perform software upgrades, and turn them on or off when necessary. RFID Middleware needs to allow users to configure, deploy and issue commands directly to readers through a common interface for example, users should be able to tell a reader when to turn off if needed.

Data Management: RFID Middleware captures EPC data from readers, it must be able to intelligently filter and route the data to the appropriate destinations. Many of the commercially available offerings take the middleware-as-a-router approach, whereas other packages support business rules, business logic, and business processes to facilitate actions and decision-making leveraging RFID data. Middleware should include both low level logic (like filtering out duplicate reads) and more complex algorithms comprehensive algorithms (content based algorithm), comprehensive solution also offer tools for aggregating and managing EPC data in either a federated (multiple repositories of data) or central data source.

Application Integration: RFID middleware solution need to provide the messaging, routing and connectivity features required to reliably integrate RFID data into existing SCM, ERP, WMS or CRM systems ideally through a service Oriented Architecture.

A service oriented Architecture is essentially a collection of services. These services Communicate with each other. The communication may involve either simple data exchange or two or more services coordinating same activity, such as order placement or shipment.

Partner Integration: Some of the most promising benefits of RFID will come from sharing RFID data with partners to improve collaborative processes like demand forecasting and Vendor managed inventory. This means that RFID middleware must provide B2B integration features like profile management, support for B2B transport protocol integration with partners data such as EDI, web based system AS2 or eventually a well engineered system specifically for EPC data.

Integration with other Auto id technologies: A complete RFID middleware offerings should be able to transform data from all types of Auto Identification and Data Capture (AIDC) input technologies, including RFID, barcode, GPS, satellite and sensors and route it to any network application. This enhancement didn't dramatically affect the prices, as it is no longer a differentiating advantages but a requirement, which reflects customer's heterogeneous RFID environments.

Centralized system: Within domain of a Centralized system, RFID Middleware can enable system to capture and store hardware and software asset information. It must be able to create detailed audit trails (with logging and reporting) to easily identify failed polling and communication sessions. It must provide hands-free maintenance, remote control and diagnostics. It must be capable enough to deploy, automate and manage anti-virus detection system.

5.2 Middleware's role within RFID solutions

The simplistic, but effective, definition of the role that RFID Middleware supports can be summed up in three primary functions:

1. Data collection and business logic at the edge of the network
2. Centralized system and device management
3. Enabling of mobile, remote and distributed systems with flexible enterprise integration.

Data collection and business logic at the edge of the network includes following(s):

- Buffering I/O activity
- EPC handling
- Event data management – cleansing, filtering, massaging data, etc.
- Event recognition – rules and exception based
- Support read/write devices
- Task management

Centralized system and device management includes following(s):

- Capture and store hardware and software asset information
- Create detailed audit trails (with logging and reporting) to easily identify failed polling or communication sessions
- Provide hands-free maintenance, remote control, and diagnostics
- Deploy, automate, and manage anti-virus detection systems

Enabling of mobile, remote and distributed systems with flexible enterprise integration includes following(s):

- Interoperability – common standards and technologies are an immediate requirement
- Manageability – remote configuration, management, and diagnosis of issues
- Scalability – suitable architectures to handle the high-volume of data in real time
- Reliability – robust systems that allow the continual flow of data
- Security – who has access to data and/or devices and who can make changes

6. Conclusion

In this chapter, we have described the basic format of the RFID data, temporal and spatial characteristics of the data. Further, we have described the problem occurred because of the parallel transponders which generates duplicate data, noisy and inaccurate data. In order to remove these inconsistencies we have discussed various filtering techniques proposed by various authors. Yu-ju-tu and Selwyn Piramuthu has proposed a mean to reduce the false positive/negative through a variation of triangulation. For triangulation purpose they consider the concept of two readers to interrogate a single tag sliding window-based algorithm is no doubt an efficient techniques but it is surrounded with various disadvantages. We have presented a scenario where raw tag data occur less than the threshold value and noise occurred more than threshold, in that case the proposed technique assume noise as correct entry and discard the real raw tag data. We have proposed dynamic updation to the threshold value and examination of EPC header.
We have also explained some challenges in data warehouse, best practices and key differences between traditional data warehousing and RFID active data warehousing. Then

we concluded the chapter with some discussion on the role of RFID middleware in data warehousing. Further, researcher may like to read the event driven and message base middleware design approach (Yulian Fei et al, 2008).

7. References

EPCglobal(2005), Tag data standards Version 1.3 standard specification" , *Auto-ID Lab*, USA

Fusheng wang and Peiya Liu(2005), Temporal Management of RFID Data, *Proceedings of the 31st VLDB Conference Trondheim,Norway*

J. Rao, S. Doraiswamy, H. Thakkar and L.S. Colby(2006), A Deferred Cleansing Method for RFID Data Analytics, *in Proceedings of the 32nd VLDB Conference,* pp. 175-186

Khan, M Ayoub et al(2009),A Survey of RFID Tags, *IJRTE* , Vol. 1, No. 4, pp. 68-71

Khan, M Ayoub, Ojha Sanjay(2009), SHA –256 based n-Bit EPC generator for RFID Tracking Simulator, In proceeding of IEEE IACC, Patiala, Punjab

M. Mealling(2003), Auto-ID Object Name Service (ONS) 1.0, *Auto–ID Lab*, MIT, USA

M. Mealling(2000), The naming authority pointer (NAPTR) DNS resource record, IETF Network Working Group Request for Comments, No 2915

M. Palmer(2004),Seven principles of effective RFID data management, http://www.progress.com/realtime/docs/articles/7principles_rfid_mg mnt.pdf,

N. Khoussainova et al(2007), Probabilistic RFID Data Management, *UW CSE Technical Report* UW-CSE-07-03-01

Oleksandr Mylyy(2006), RFID Data Management, Aggregation and Filtering, *Hasso lattner Institute at the university of Potsdam*, Germany

R. Derakhshan, M. Orlowska, and X. Li (2007), RFID data management: Challenges and opportunities, *Proc. IEEE International Conference on RFID 2007*, pp.175–182

Auto-ID Technical report(2002), *860MHz–930MHz EPC Class I, Generation 2 RFID Tag & Logical Communication Interface Specification*, Auto-ID Centre

Venkat Krishnamurthy et al(2004), Managing RFID Data , *Proceedings of the 30th VLDB conference, Toronto*, Canada

Y. Bai et al (2006),Efficiently filtering RFID data streams, Proc. CleanDB Workshop, pp.50–57

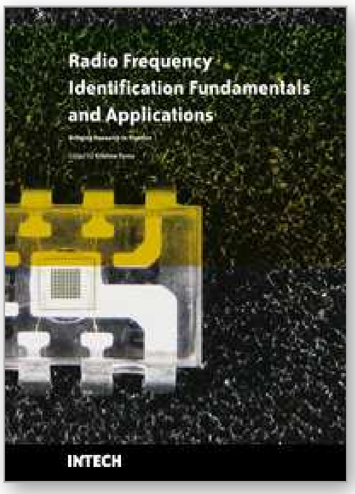
Yulian Fei, Gonglian Jin, Ruqi Wu(2008), Research on Data Processing in RFID Middleware Based on Event-Driven, IEEE International Conference on e-Business Engineering, pp.578-581

Yu-Ju-Tu and Selwyn Piramuthu(2008),A Reducing False Reads in RFID-Embedded Supply Chains, Journal of Theoretical and Applied Electronic Commerce Research ISSN 0718–1876 Electronic version Vol. 3 Issue 2 pp. 60-70

Hector Gonzale , jiawei Han , Diagoe Klabzan(2006), Warehousing and Anaylysing Massive RFID Datasets, *ICDE* 2006

Yu-Ju-Tu et al(2008), A décision support model for filtering RFID Data,*IEEE ADCOM*

Yijian Bai, Fusheng Wang, Peiya liu, Carlo Zaniolo, Shaorong Liu,(2005) , RFID Data Processing with Data stream Query Language , Siemens Corporate Research



Radio Frequency Identification Fundamentals and Applications Bringing Research to Practice
Edited by Cristina Turcu

ISBN 978-953-7619-73-2
Hard cover, 278 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

The number of different applications for RFID systems is increasing each year and various research directions have been developed to improve the performance of these systems. With this book InTech continues a series of publications dedicated to the latest research results in the RFID field, supporting the further development of RFID. One of the best ways of documenting within the domain of RFID technology is to analyze and learn from those who have trodden the RFID path. This book is a very rich collection of articles written by researchers, teachers, engineers, and professionals with a strong background in the RFID area.