

# RFID Event Analysis Based on Complex Event Processing

<http://dx.doi.org/10.3991/ijoe.v10i1.3049>

Kong Xiangsheng

Xin Xiang University, Xin Xiang, China

**Abstract**—A variety of devices and ubiquitous computing systems such as RFID (Radio Frequency Identification) generate and exchange significant events during their execution. In this paper, compared to most existing RFID framework, We provide a framework of RFID system based on Complex Event Processing which aims at providing a scalable solution for gathering, filtering, and providing clean RFID data. We design a complex event detection algorithm of performing Complex Event Processing directly over unreliable RFID event streams by incorporating cleansing requirements into complex event specifications. We explain how raw RFID events can make business process data accessible for process mining. The results of this research can be widely used for RFID events analysis in various areas such as logistics and manufacturing systems.

**Index Terms**—Esper, RFID, Complex Event Processing, complex event detection

## I. INTRODUCTION

More and more enterprise applications using Radio Frequency Identification (RFID) technique introduce a need for an infrastructure that hides proprietary device interfaces, facilitates configuration and monitoring of the devices, and processes the captured data. Now RFID is rapidly expanding their applications to many areas: inventory systems, supply chains, access control, vehicle tracking, toll payments, e-ticketing, pharmaceuticals and so on. An RFID system consists of three basic components: a tag (containing a microchip and an antenna which ensures the communication with a dedicated reader), a reader (or a sensor) and a back-end server. The reader sends an RF signal which can power the receiver (the tag). This in turn will radiate back a signal to the reader. This backscattered signal usually contains a number that uniquely identifies the tag/item [1]. The back-end server mainly aids the reader by an extra storage spaces and several services (such as filtering, fusion, aggregation, and correlation of events) that perform real time processing of tag event data and collect the identifier of objects interrogated by RFID readers at different time points and locations. That extra storage space can be used to keep the information of all tags that can be accessed by the reader and the information of the results.

Raw data present little value until they are transformed into a form, known as semantic based RFID complex event, which is suitable for application-level interactions. Therefore, from an application perspective, it is desirable to provide a mechanism that turns the low-level captured data into the corresponding business event. For example, a detection of a number of tagged books at the exit door of a

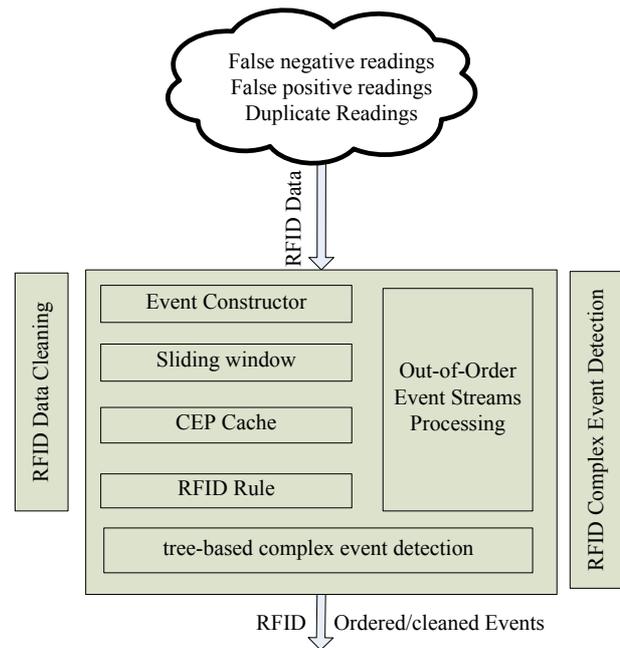


Figure 1. the framework of RFID system based on Complex Event Processing

library can be automatically translated into a books checked out event, which is more expressive to meet the requirement of RFID applications.

Traditional RFID solutions spend much effort on raw event processing instead of business level, actionable event. And they are not sufficient to provide dynamic invocation of logistic services. According to [2, 3] an event is a record of an activity in a system and may be related to other events. Complex Event Processing has become one of the most promising research areas in the domain of event management now. Especially, in logistic environments with numerous RFID events which should be related to proper logistic services. By the use of Complex Event Processing engines, low-level events can be aggregated to high level events in real time [4]. The purpose of this research is to develop an event processor that detects sequence events and design a Complex Event Processing framework that is capable of detecting complex issues.

In the light of RFID readers' expressive limitation, there is an expression gap between expression of primitive events and the real world requirement. The paper is composed as follows. The framework of RFID process mining is presented in Section 2. To illustrate the proposed framework, an algorithm of RFID base on Complex Event Processing is introduced in Section 3. An example of

RFID process mining is explained with some process mining techniques with the implemented system in Section 4. Finally, Section 5 concludes this paper.

## II. FRAMEWORK OF RFID SYSTEM BASED ON COMPLEX EVENT PROCESSING

The primary differences between the RFID technology and others lie in the real-time events detection and automatic information gathering without human interaction. RFID technology has dramatically lowered the cost of data acquisition. The framework of RFID system based on Complex Event Processing aims at providing a scalable solution for gathering, filtering, and providing clean RFID data to the end-user (shown in Fig. 1). The first stage is to collect and process events which are generated from RFID readers. The next stage is to combine real-time events in the message queue with past events in the database. The final stage is to analyze transformation events by process mining algorithms [5]. There are two important issues in transforming low-level observations into high-level events. The first issue is RFID data cleaning. The second issue is RFID complex event detection.

### A. RFID Data Cleaning

The noisy, low-level data produced by RFID readers is almost impossible to use or comprehend in most but the simplest settings. These devices generate massive sensor data streams, and a variety of primitive physical events can be generated by analyzing them. Since sensor data streams are continually generated, the primitive events are also generated continuously. The continual events can be considered a sequence of primitive events, and recognition of the sequence provides us a high level event, which is easy to understand in the physical world. Generally, these unreliable readings can be formalized in three typical undesired scenarios: false negative readings, false positive readings and duplicate readings, discussed as follows.

**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.

**False negative readings—**RFID data contains significant amounts of false negatives. In general, error rates depend on the equipment used, the object material (e.g., metallic objects or objects containing water have much lower detection rates than other objects), and the orientation of tags and antennas. In this case, RFID tags, while present to a reader, might not be read by the reader at all.

**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.

**False positive readings:** RFID data may also contain false positives where nearby antennas detect the same tag at the same time. In this case, besides RFID tags to be read, additional unexpected readings are generated.

**Duplicate Readings—**Duplicate observations are very common in RFID system when multiple readers send signals to a particular item at the same time.

To improve data quality and reader reliability we introduce RFID complex event detection to filter out abnormal and corrupted readings, remove duplicate readings, and smooth readings to assist in recreating missing data from imperfect readers.

### B. Sliding Window Technique

Sliding is the process of interpolating to compensate for lost readings and discarding anomalous readings by running a windowed aggregate (in this case, a count) over the cleaned stream. A sliding window is a window with certain size that moves with time. This window is used to define this time interval. Suppose the window size has time coordinate of  $[t_1, t_1 + \text{window\_size}]$ , after  $s$  time, the coordinate will become  $[t_1 + s, t_1 + \text{window\_size} + s]$ . RFID readings in form of tuples (ReaderID, ObjectID, TimeStamp, ArrivalTime) will enter the sliding window and get expired as time moves. We use the sliding time on the same RFID reader. Solid circle represents the label that has been read, and hollow circle represents the label that has not been read. If  $\langle O, pr_1, t_1 \rangle, \langle O, pr_1, t_3 \rangle$ , and  $t_1 < t_2 < t_3$  result is true, it can be speculated that  $\langle O, pr_1, t_2 \rangle$  should also be included in the read set, and it can be added to the output data stream after the sliding time.

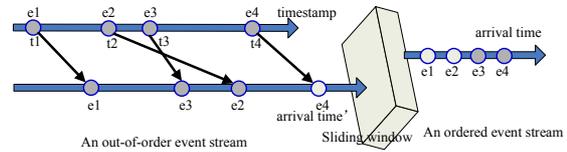


Figure 2. Out-of-order Event Stream

### C. Out-of-Order Event Streams Processing

Let  $e_i$  and  $e_i.ts$  be the arrival timestamp and the occurrence timestamp of an event  $e_i$ . Consider an event stream  $S: e_1, e_2, \dots, e_n$ , where  $e_1.ts < e_2.ts < \dots < e_n.ts$ . For any two events  $e_i$  and  $e_j$  ( $1 \leq i, j \leq n$ ) from  $S$ , if  $e_i.ts < e_j.ts$  and  $e_i.ts < e_j.ts$ , we say the stream is an ordered event stream. If however  $e_j.ts < e_i.ts$  and  $e_j.ts > e_i.ts$ , then  $e_j$  is an out-of-order event. In the example of Fig. 2, the timestamps of events  $e_1$ -- $e_4$  are listed in order. But we can see that event  $e_2$  arrives later than event  $e_3$ , which is called out-of-order [6].

The most common approach on handling out-of-order event streams is to use K-Slack as an a priori bound on the out-of-order input streams. It buffers incoming events in the input queue for  $K$  time units until the order can be guaranteed. The biggest drawback of K-Slack is the rigidity of  $K$ , which cannot adapt to the variance in the network latencies that exist in a heterogeneous RFID reader network. To overcome above shortcoming, we propose a strategy which improves K-slack according to practical application.

One way to deal with the gush of RFID data is to develop RFID data "concentrators" that help buffer the flow of RFID event streams. We use Adaptive Learning Algorithm to adjust time units, current time is compared to the tag captured time, accordingly, the delay time of current data is obtained, and we use  $K$  time units as a standard to estimate the delay time of next time node, in order to adjust time for cache dynamically, and improve agility and adaptability of this algorithm.

The model adopts divide strategy, namely, different types of events in the system will be processed differently. Every event can trigger correlation actions according to rules worked out by Rule Configuration Module.

#### D. RFID Rule

Associating relationship among objects has been identified as a difficult issue for RFID applications [7]. This problem can be solved with this rulebased approach. In [8] is shown how declarative RFID rules can provide powerful support for event processing, including event filtering, event transformation and aggregation, and realtime monitoring. RFID rules can be defined using the following syntax:

```
Rule (rule_id, rule_name)
ON {(object).(event)}
IF conditionSet={(condition_expression)}
Do actionSet={(object).(action)}
End Rule
```

where conditionSet is a boolean combination of event state including Not ready, Ready, Running, Waiting, Completed, Failed and Aborted etc. ActionSet is an ordered list of actions including Initiate(), Start(), Suspend(), Resume(), Reject(), Fail(), Abort(), Finish() and SendAlert() etc. For example:

```
Rule (20121215001, initiate_rule)
ON {CE1.detected}
IF conditionSet={RFIDReader.state= = Not ready}
Do actionSet={RFIDReader.initiate();RFIDReader.state= = Ready}
End rule
Rule (20121215002, begin_rule)
ON {CE2.detected}
IF {SEQ(CE1, CE2), RFIDReader.state= =Ready}
Do { RFIDReader.start(),RFIDReader.state= = running}
End Rule
```

#### E. RFID Complex Event Detection

**Definition 1:** RFID primitive event. A RFID primitive event is an observation read by the RFID reader from tags. It is the basic unit of events. It is defined as:

(ReaderID, ObjectID, TimeStamp, ArrivalTime)

This is a four-tuple, which contains the following elements: the ID of reader, the ID of tag, the time stamp of the occurrence of event and the arrival time[9]. TimeStamp records the time that the event providers generate the events. ArrivalTime is the time that events arrive at the back-end server (responsible for processing the event).

**Definition 2:** RFID complex event: A RFID complex event is the output event of Complex Event Processing engine. During the processing period, Complex Event Processing engine will operate upon RFID basic events according to specified user definition or real-time application requirement.

**Definition 3:** RFID Event Expression and Event Operator. In this paper, a composite event is defined by applying an event operator(shown in Tab.1) to constituent events that are primitive or other composite events.

For example if the complex event CE1 (or CE2) is formed by three primitive events PE1, PE2, PE3 (or PE1, PE2, PE3) captured by a reader, the format is like this:

CE1 { ReaderID, (PE1  $\wedge$  PE2  $\wedge$  PE3), Timestamp, ArrivalTime }

CE2 { ReaderID, SEQ(PE4,PE5,PE6), Timestamp, ArrivalTime }

In the absence of event operators, several rules are required to specify a composite event. Furthermore, some control information needs to be made a part of a rule specification.

TABLE I.  
RFID EVENT EXPRESSION AND EVENT OPERATOR

Operator	Example	Meaning
$\wedge$	$E_1 \wedge E_2$	Both $E_1$ and $E_2$ occur without occurrence order
$\vee$	$E_1 \vee E_2$	Either $E_1$ or $E_2$ occurs without occurrence order
$\sim$	$\sim E$	E does not occur
SEQ	SEQ( $E_1, E_2$ )	Sequence of $E_1$ and $E_2$ , $E_2$ occurs under the condition that $E_1$ has already occurred
TSEQ	TSEQ( $E_1, E_2, t_{e1}, t_{e2}$ )	Based on the sequence of $E_1$ and $E_2$ , calculate the time distance between those two events
WITHIN	WITHIN(SEQ( $E_1, E_2$ ))	WITHIN operator specifies that an instance of $E_2$ needs to happen after an instance of $E_1$ but within t from it

After the application logic was interpreted as complex events, we should consider an effective algorithm for RFID event detection. RFID Complex Event Detection Algorithm was proposed in [10]. We use the tree-based structure to detect RFID complex events. The main idea of the tree-based complex event detection approach is to construct matching trees from complex event mode. In a matching tree, leaf nodes represent primitive events and other nodes denote the composition of events. Fig.3 demonstrates an example of graphical representation of complex events: WITHIN(TSEQ (E1 AND E2, 0.1sec, 1 sec); E3, 10 min).

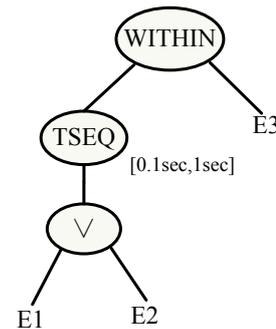


Figure 3. An Example of Graphical Representation of Complex Events

RFID primitive events are predefined in the RFID system and are detected at the time of occurrence. RFID composite events compose of more than one event and it can be either a primitive event or a composite event itself. Thus, RFID composite events can be detected at the time when the last constituent of an event expression occurs or over an interval. RFID composite event detection involves two steps: (1) checking the detection condition based on the operator semantics, and (2) time of detection[11]. In applications where event detection along a sliding time window is needed, continuous context can be used. This

context is especially useful for tracking trends of interest on a sliding time point governed by the initiator event.

### III. ALGORITHM OF RFID SYSTEM BASED ON COMPLEX EVENT PROCESSING

We show a baseline implementation of denoising as shown in Algorithm 1. The input of the algorithm RFID system based on Complex Event Processing: RFID rules set:  $R=\{r_1,r_2,\dots,r_m\}$ , primitive event set:  $E=\{e_1,e_2,\dots,e_n\}$ . Output: The set of detection result  $H$ . At the beginning of algorithm, we construct event tree  $T$  according to the RFID rule  $r$  in set  $R$ , then we initiates  $T$  by the following steps.

- (1) Propagate interval constraints from the root node of  $T$  in a top-down way.
- (2) Set counters for every non-leaf node. And the number of counters depends on the number of child nodes. These counters increases when the node is activated, while decreases when a match occur.
- (3) Build node event queues for each leaf node in  $T$ .
- (4) Assign flags for each non-spontaneous node, and build expected event queues for them.

```

Algorithm 1 Baseline denoise (params: window size, threshold)
WINDOWBUFFER ← empty queue {FIFO queue to hold sliding window of readings}
loop {loop forever for next incoming reading}
  INCOMING ← the next reading
  append INCOMING to the end of WINDOWBUFFER
  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 WINDOWBUFFER 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
    
```

### IV. PERFORMANCE STUDY

For experiments, a random RFID reading generator was created, which generates RFID tag reading according to a Poisson process. The Poisson process generates tag readings with random arrival time, while the arrival time conforms to a Poisson distribution with a chosen average tag arrival rate. Each generated tag reading repeats for 10 times, with some chosen noise level (a certain percentage of the reading are noise). In the environment, our algorithm still achieves the shortest tag identification time (shown in Fig.4). In this environment, however, we observe performance gaps between the time-optimal and the analytical values as the tag number is increased. Our analysis on this phenomenon has found that the gaps come from the accumulated estimation error related to strong limitation of frame size.

### V. CONCLUSIONS AND FUTURE WORK

In this paper, we provide an RFID event-based triggering method of logistic processes managed by a BPM system. This event notification mechanism is based on the improved ECA rules. And our study proposes a framework to merge both RFID event processing and logistic process management that can implement ubiquitous logistic process management system. Complex Event Processing engine could extract the time, causal, hierarchical and other relationship among RFID events based on user's defined rule. Thus it could discover more high level, actionable information behind primitive events. These business level events could be regarded as the trigger of a new internal or external activity, invoke or change those event-driven activities.

There are several other opportunities for future work as well. First, a practical comparison of the presented papers, solutions, methods, and tools should be performed by using or defining a benchmark designed for this purpose. A practical comparison could show the concrete differences between the different tools in terms of speed, memory consumption, accuracy, and other aspects. Another future direction could be to participate in the development of an open source Complex Event Processing solution, e.g., Esper, thus gaining valuable practical experience. Although this survey tries to comprehensively overview the ongoing research results of the event processing research area, it could be extended with real experiments coming from the development of practical Complex Event Processing applications.

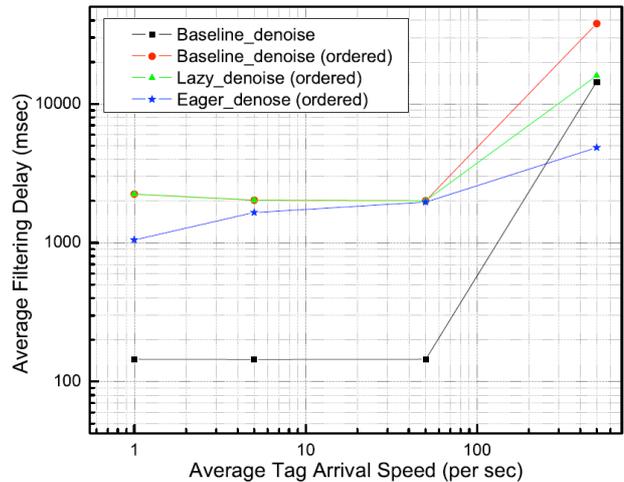


Figure 4. Duplicate Elimination: Delay under Different Arrival Rates

### REFERENCES

- [1] Oleksandr Mylyy, "RFID Data Management, Aggregation and Filtering", Proceedings of the 31st International Conference on Very Large IAJIT First Online Publication Data Bases Seminar on RFID Technology, 2007, pp. 6–154.
- [2] Xue Yinghua and Liu Hongpeng, "RFID-based intelligent storage and retrieval systems in automated warehouse", Journal of Software, Vol 6, Issue 9, 2011, pp. 1844-1850.
- [3] M. Safkhani and M. Naderi, "Cryptanalysis and Improvement of a Lightweight Mutual Authentication Protocol for RFID system", 7th International ISC Conference on Information Security and Cryptology, 2010, pp. 57–59.
- [4] Songsen Yu, Yun Peng, Jian Yang and Jiajing Zhang, "The design and realization of a lightweight RFID mechanism integrating secu-

SPECIAL FOCUS PAPER  
RFID EVENT ANALYSIS BASED ON COMPLEX EVENT PROCESSING

- rity and anti-collision", *Journal of Software*, Vol 6, Issue 7, 2011, pp. 1235-1240.
- [5] Bai Kun, Li Shaoxi, Liu Tao, Tian Jing, "A novel method for modeling RFID data", *Journal of Software*, Vol 7, Issue 4, 2012, pp. 835-843
- [6] Yijian Bai, Fusheng Wang and Peiya Liu, "Efficiently Filtering RFID Data Streams", In *CleanDB Workshop*, 2006, pp. 50-57.
- [7] Jeffery, S., Garofalakis, M. and Franklin, M., "Adaptive Cleaning for RFID Large Data Bases", *Proceeding of the 32nd international conference on VLDB*, 2006, pp. 163-174.
- [8] Zhang Xiaobo, Cheng Lianglun and Zhu Quanmin, "Improvement of filtering algorithm for RFID middleware using KDB-tree query index", *Journal of Software*, Vol 6, Issue 12, 2012, pp. 2521-2527
- [9] Chunjie Zhoua, Xiaofeng Menga and Yueguo Chenb, "Out-of-order durable event processing in integrated wireless networks", *Pervasive and Mobile Computing*, vol. 7, Issue 5, 2011, pp. 595-610. <http://dx.doi.org/10.1016/j.pmcj.2010.11.010>
- [10] Minbo Li and Hua Li, "Research on RFID integration middleware for enterprise information system", *Journal of Software*, Vol 6, Issue 2, 2011, pp. 167-174
- [11] Hongying LIU, Satoshi GOTO, and Junhuai LI, "The Study and Application of Tree-based RFID Complex Event Detection Algorithm", *Proceedings of the 2009 International Symposium on Web Information Systems and Applications*, 2009, pp. 520-524.

AUTHORS

**Kong Xiangsheng** is a Research Scholar in the Department of Computer Engineering, Xinxiang University, Xinxiang, China (e-mail: fallsoft@163.com). He received B.E. Degree in 2003 from Information Engineering University, China. He has teaching experience of 10 years. He has published one PHP book and two MySQL books and four international papers. His interests are in system analysis & design and software testing.

Submitted 24 July 2013. Published as re-submitted by the author 23 January 2014.