PTSA Event Scheduling Algorithm of CEP-Based RFID Tracking and Locating System

Guangqian Zhang
The institute of system engineering and researching
Dalian University of Technology
Liaoning, China
zhgq@dlut.edu.cn

Li Zhang, Xin Liu
The institute of system engineering and researching
Dalian University of Technology
Liaoning, China
realm@i26.com

Abstract—RFID-based tracking and locating system faces many challenges. In order to solve the problem, we build a model of RFID-based tracking and locating system which is based on complex event processing, that is BCEPS model. BCEPS model consists of physical device layer, RFID data processing layer, event processing engine, application layer, data storage layer. BCEPS model filters the RFID data which is gathered in real time and then generates RFID events. At the same time, BCEPS model responds to the requirement coming from users and then generates user events. RFID and user events are sent to event processing engine and are processed. Finally, results are returned to users. To enhance the efficiency of processing massive events and the real-time performance of system, we first propose and design the real-time event scheduling algorithm for event processing engine. Finally, we verify BCEPS model by implementing RFID-based tracking and locating system in underground mine.

Keywords—RFID; PTSA event scheduling; complex event processing; tracking and locating

I. INTRODUCTION

At present, RFID technology has been applied for mobile commerce [21], pervasive computing [22]. For example, m-commerce based on NFC technology is the typical application which applies RFID to mobile business. RFID technology not only stores the information about objects, but also holds the promise of real-time identifying, locating and monitoring physical objects. Combined with information technology, RFID can provide location-based services which are fundamental and critical for context awareness service [23].

However, RFID used to track and locate objects faces the following challenges: 1) RFID reader could gather large quantity of primitive data that causes numerous primitive events. Primitive data may be duplicated and in conflict each other. So how to process conflicted data and filter duplicated data to avoid generating invalid or conflicted event is important to save storage spaces and improve performance of system; 2) because of the limitation of RFID reader’s expression and logic complexity of applications, RFID reader only generate the primitive event that can not meet the real world requirement. However, applications require meaningful and actionable information. Thus, how to transform RFID data and generate complex event meeting the actual application requirement is important for functionality of system. 3) During the process of RFID readers gathering RFID tags, many kinds of events containing real-time and non-real-time events will be generated. Scheduling events unreasonably and ineffectively will affect real time of BCEPS system. So, it is important for BCEPS system to schedule events reasonably and effectively.

It is proved that the newly complex event processing system has solved the former two challenges discussed above. Some CEP systems such as SASE [2] have been designed for RFID information transformation. The third challenge can be solved by designing a real-time event scheduling algorithm.

Currently, the research on complex event processing in RFID mainly focuses on RFID data cleaning [19], RFID event detection [20] and the design of complex event processing language [2]. However, RFID devices will generate massive RFID data and events and cause “events flood”. If events are scheduled unreasonably and disorderly, the results and real time of system will be affected heavily.

In this paper, we propose the BCEPS model system which is based on complex event processing, BCEPS model system a general tracking and locating model and uses PTSA event scheduling algorithm that we design by ourselves to enhance real time.

This paper is organized as follows. Section II presents the definition of RFID primitive data stream and RFID events. Section III presents the definition of user events and complex event processing language. Section IV presents the architecture of BCEPS model system. Section V presents the PTSA event scheduling algorithm. The results of the experiment that was conducted are presented in section VI. Finally, the conclusion and future works are presented in section VII.

II. RFID EVENT DEFINITION

A. RFID Primitive Data Stream

RFID primitive data streams gathered by RFID readers are independent and primitive information. RFID primitive data streams’ formats are <r,o,t> which represent RFID reader r gathers RFID Tag o at time t. Therefore, RFID primitive data streams need to be transformed into meaningful and actionable RFID events, illustrated by Figure 1.
B. RFID Event

RFID events have concrete meanings and can be processed by model system directly. The RFID event format is like:

\[ \text{Erfid (reader id, tag id, timestamp)} \]

Where reader id represents the reader which exists in the event; tag id represents the tag which exists in the event; timestamp represents the time when RFID reader gathers the RFID tag. The event represents that RFID reader whose number is reader id gathers the RFID tag whose number is tag id at the time timestamp.

III. COMPLEX EVENT DEFINITION

A. User Event

Complex events represent the complex requisitions which come from users, called user events. User events can not be processed directly, so they should be transformed into several meta-events which can be processed directly. User events format is like:

\[ \text{Euser (CEP, rules, timestamp)} \]

Where CEP represents a complex event which comes from user’s requisition; rules are used to transform CEP into meta-events; timestamp represents the time when the user event is generated.

Several rules by which complex events are transformed into meta-events are defined in advance. Complex events are transformed into several meta-events by complex event processing language, illustrated by the part of Figure 2 which is surrounded by the dotted line. Model system processes meta-events and return results of meta-events. Finally, the result which meets requirement of the complex event is generated by combining these results of meta-events, illustrated by the part of Figure 2 which is surrounded by the real line.

B. Complex Event Processing Language

Complex event processing language can transform user events (complex events) into meta-events and return final results which meet the requirement of user events. Complex event processing language makes the transformation between user events and meta-events transparent and convenient. The grammar of complex event processing language is similar to SQL and its format is like:

\[
\text{SELECT [result] FROM [meta-events] LIKE [CEP] WHERE [condition]}
\]

Where SELECT clause sends final result into result; FROM clause specifies the range of meta-events which is related to to CEP, if not declared, default range of meta-events is used; LIKE clause represents the complex event to be processed, WHERE clause represents the extra conditions which come from users.

IV. BCEPS MODEL

We design and construct tracking and locating model system which is based on complex event processing, called BCEPS model system. Figure 3 illustrates the architecture of BCEPS model system.

BCEPS model system consists of physical device layer, RFID data stream processing layer, event processing engine, application layer and data storage layer. In physical device layer, RFID readers gather RFID tags which hold real-time information and generate RFID primitive data stream which will be sent to RFID data stream processing layer finally. RFID data stream processing layer filters RFID primitive data stream to eliminate redundant, duplicate and abnormal data streams and produces RFID events which will be sent to event processing engine. At the same time, application layer yields several users’ requisitions which will be encapsulated in user events and sent to event processing engine. All kinds of events received by event processing engine are stored in a temporal buffer called event pools. Event processing system in event processing engine processes the chosen event and returns the final result to users. The details of BCEPS model system are as follows.
1) Physical device layer
The layer consists of RFID readers and RFID tags. RFID readers gather RFID tags in real time.

2) RFID data stream processing layer
This layer consists of event filtering layer and event generating layer, illustrated by Figure 4. Event filtering layer eliminates duplicate, redundant and abnormal RFID primitive data streams generated by physical device layer. Event generating layer yields the RFID events which are constructed from filtered primitive data streams.

Figure 4 demonstrates the process of RFID data stream processing layer. At first there are five RFID primitive data. The first data is an invalid data because it loses the timestamp. The second data and the third data are duplicated. And then, the five RFID primitive data are processed by event filtering layer and there are three RFID primitive left which are valid. Finally, three RFID events are generated by event generating layer.

3) Event processing engine
Event pools and event processing system constitute event processing engine. Event pools are a temporary space to store all events to be processed. There are two kinds of events including RFID events and user events.

Event processing engine chooses the most superior event according to some real-time event scheduling algorithm which will be introduced in the next section. Event processing system processes events in different ways in the light of the event type. In regard to RFID events, event processing system processes RFID events by the algorithm dependent on specified application to update real-time information. In regard to user events, event processing system processes meta-events in multithread technology and returns the results meeting the requirements of users to application layer.

4) Application layer
The layer provides users with convenient interface to produce appropriate user events in accordance with users’ requirements. And it shows the final results returned by event processing engine in the way of graphical mode.

5) Data storage layer
The layer stores the basic data and rules which are used by event processing engine.

V. REAL-TIME EVENT SCHEDULING ALGORITHM
Event processing engine of BCEPS model system receives two kinds of events including RFID events and user events. RFID events are the foundation of BCEPS model system. The results of user events are obtained by analyzing and processing the information which comes from RFID events. If event processing engine processes events only according to the time when the events arrive at the event pools, it will happen that users get the results which are inconsistent with the reality. In order to guarantee that the results are right and real time, we propose PTSA event scheduling algorithm based on priority and timestamp.

A. Event Definition
We encapsulate RFID events and user events in a general format. The format is like:

EVENT (type, Etype, priority, timestamp)

where type represents the type of the event which refers to rfid and user; Etype represents the content of the event which is in detail introduced in the Section 2 and 3; priority represents the priority of the event, the bigger the value of the priority is, the higher the priority of the event is; timestamp is not the time when the event arrives at event pools but the time when the event should be completed. For RFID events, timestamp is equal to the timestamp of Efid. For user events, if the CEP of Euser includes timestamp, timestamp is equal to the timestamp of CEP of Euser. If the CEP of Euser does not refer to timestamp, timestamp is equal to the timestamp of Euser. For example, Euser(‘query the location of miner A at time 111’, rules, 112) represents a user event and the timestamp of EVENT(user, Euser, 3, 112) is equal to 111. Euser(‘get the list of miners’, rules, 112) represents the other user event and the timestamp of EVENT(user, Euser, 3, 112) is equal to 112.

B. Organizational Structure and Classification of Event
Events in event pools are organized in terms of two different structures. One organizational structure is FIFO event queue which is according to the time when the events arrive at event pools. The other organizational structure is priority-based event queue which is according to the priorities and timestamp of events. Figure 5 illustrates the two kinds of organizational structures. The event pools in the
Figure 5 have eight events. Figure 5 (a) represents the FIFO event queue and Figure 5 (b) represents the priority-based event queue.

From the perspective of real-time, events include hard real-time events and soft real-time events. The priority of hard real-time events is always higher than soft real-time events. Soft real-time events could be preempted by hard real-time events. Currently PTSA event scheduling algorithm does not refer to hard and soft real-time events scheduling.

C. PTSA Event Scheduling Algorithm

The process of PTSA event scheduling algorithm is as follows:

1. Choose the earliest event to be processed as EVENT A which arrives at event pools.
2. If EVENT A is RFID event, it will be sent to event processing system and PTSA algorithm finishes.
3. If EVENT A is user event, pick out RFID events whose priorities are bigger than EVENT A’s priority and timestamps are prior to EVENT A’s timestamp, called Erfid sets.
4. Process Erfid sets and then process EVENT A.
5. PTSA algorithm finishes.

D. Complex Event Processing Mechanism Based On Multithread

User events must be transformed into several meta-events processed directly by model system. Some meta-events have interrelationship, but others are independent, illustrated by Figure 6. Meta-event 2 and 3 require the output of meta-event 1 as input. However, there is not interrelationship between meta-event 2 and 3. Therefore, we process independent meta-events using multithread technology to enhance real-time of model system and improve responding time of model system.

VI. EXPERIENCE

The proposed BCEPS model system was tried in the following environment. Due to the restriction on the organization of an environment with mobile business, it was simulated with RFID-based tracking and locating system underground mine (RTLUM). RTLUM was building based on BCEPS model system. RTLUM consisted of data collection module and graphic monitoring module. Data collection module gathered real-time position information about miners tied with RFID tags, filtered RFID real-time position information, generated RFID events and sent RFID events to event processing engine. Graphic monitoring module generated user events and showed the results graphically which came from event processing engine. We used time shifting tracking and locating algorithm
table

Figure 7 illustrates event pools at time t. At time t, the earliest event is Erfid(reader 1, tag 1, 111). According to PTSA algorithm, it will be processing directly. Figure 8 illustrates event pools at time t+1. At time t+1, the earliest event is Euser(query the information of 'Robby' with tag 2, 114). Because some RFID events referring to tag 2 and their timestamps are prior to the user event, these RFID events should be processed prior to the user event. Therefore, at time t+1, Erfid(reader 1, tag 2, 112) should be processed.

Figure 9 illustrates the scene of processing user events using multithread technology.
VII. CONCLUSION

This paper proposes BCEPS model system which is based on complex event processing. BCEPS model system involves two kinds of events: RFID events and user events and gives the detailed definitions of them. Then we describe the architecture and components of BCEPS model system. In order to improve responding time and enhance real time, we propose PTSA event scheduling algorithm and the mechanism of processing user events using multithread technology. Currently PTSA event scheduling algorithm does not involve hard and soft real-time events scheduling. In the future, we will study further on scheduling hard and soft real-time events.

ACKNOWLEDGMENT

This work is supported by the doctoral fund of the Ministry of Education (20070141037) and the National High Technology Research and Development Program of China (863 Program: 2007AA04Z103).