International Journal of Intelligence Science, **2012, 2, 160-165**
http://dx.doi.org/10.4236/ijis.2012.224021 Published Online October 2012 (http://www.SciRP.org/journal/ijis)

RFID Complex Event Processing: Applications in Real-Time Locating System

Yao-Zong Liu, Hong Zhang, Yong-Li Wang
School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China
Email: new025@126.com

Received June 29, 2012; revised August 15, 2012; accepted August 24, 2012

ABSTRACT

Complex event processing (CEP) can extract meaningful events for real-time locating system (RTLS) applications. To identify complex event accurately in RTLS, we propose a new RFID complex event processing method GEEP, which is based on the timed automata (TA) theory. By devising RFID locating application into complex events, we model the timing diagram of RFID data streams based on the TA. We optimize the constraint of the event streams and propose a novel method to derive the constraint between objects, as well as the constraint between object and location. Experiments prove the proposed method reduces the cost of RFID complex event processing, and improves the efficiency of the RTLS.

Keywords: Complex Event Processing (CEP); Real-Time Locating System (RTLS); Radio Frequency Identification (RFID); Timed Automata (TA); Event-Clock Automata (ECA)

1. Introduction

In recent years, radio frequency identification (RFID) is more and more used in positioning, tracking and analysis system for moving objects. It reads the information of mobile tags within the effective range by fixed readers to identify and locate moving objects [1]. Real-time locating system (RTLS) uses RFID real-time locating to identify moving objects with the label, and makes appropriate treatment based on location and other information [2]. Events in the positioning system are complex event (CE) consisting of primitive events (PE). Wen Yao [3] propose a CEP framework to model surgical events and critical situations in a RFID-enabled hospital.

With the gradually increasing in RFID applications (such as network monitoring, supply chain management, and moving objects tracking, etc.) needing to handle large volumes of RFID data streams [3], the users can describe the required detection mode (complex event) by the complex event definition language, so research of RFID complex event detection mechanism for a specific application is increasingly important. (Complex Event Processing, CEP) system [4] must have a quick response to incidents existing data analysis tools gain knowledge inherent in the data by running complex queries against the database, they can only query the data of the past, but unable to understand current events. Currently, there is no general data-processing tools can easily respond to events in time in real-time occasions, timed automata is a very useful tool to solve the synthesis of RFID complex events [1].

To solve problems which RFID for RTLS encounters in complex event processing, this paper proposes a method for RFID complex event processing based on timed automata theory, this method can make real-time process for complex events in RFID data streams.

2. Related Works

David Luckham [5] gave definition of complex events in the 2001 book “The Power of Events”: complex event detection is a pre-defined collection of tools and techniques designed for analysis and control of a series of interrelated events.

Present event definition language regards events as transient, although some events occur on a time period. They all use semantics based on detection. They can only detect occurrence of composite events by the end time of events, but cannot distinguish detection of events from occurrence of events. So they have significant limitations for those applications based on period semantic, for instance, in the detection of the timing events, operations in the events appear in different combinations will lead to inconsistent semantics.

The core issue of event-centric approach is complex event processing (Complex Event Processing, CEP), namely, how to efficiently make combination of primitive events to be complex events with more complex semantics, including considering the restrictions between various events, even in some applications need to continue to detect complex events to generate a higher level of complex events. The following methods are available: Finite Automata models, Petri Net models, Match Tree models and models based on graph [1].

Complex event system for data stream is mainly responsible for the detection of complex events, Primitive events can be connected to constitute a complex event by "logical" timing operator. A language used for filtering and associating events is needed for modeling complex events. Many papers have studied the complex event language, such as publish/subscribe [6], data stream processing, complex event language, etc., but languages have been proposed have significant defects in the detection of RFID event. Here we take shoplifting as an example to explain this defect. Shoplifting is consisted by a series of events describing the scene, picking up a product from the shelves, taking out of the store without checking. From the example, we should address some problems, for instance, if the event actually happened, adding time constraints (sliding window queries) and value-based constraints for the event. WU [7] has proposed a query language (SASE) to support the complex event processing.

To achieve the complex event language, we need the support of the data structure in the underlying language model. There are many researchers in different areas using different data structures, XML filtering, publish/order, active database extension. They can be divided into three categories: automata-based, Petrinet based, tree and graph-based approach. Altinel [8] proposed X-filter using finite state machine (FSM), for each query, the index method allows the simultaneous execution of all FSMs in processing of documents. Using SAX [9] to code every document into events sequence as X-filter system input. Diao [10] proposed Y-filter, namely, combining all FSMs to a non-deterministic finite automata (NFA) to enhance the time and space using performance of X-filter. For complex event detection, an effective method is using non-deterministic finite state automata (Nondeterministic Finite Automata, NFA) to represent the structure of an event sequence. In addition, non-deterministic finite state automata-based approach combining with the extension of the XML message filtering technology can be used to process the reconstruction of events by sequence.

In SASE [7], researchers extend the model Y-filter, they process the complex event query on the data stream of real-time RFID reading. The key concept time is ignored in the SASE. In many RFID applications, an event is considered to be an effective event, if it happens within the time limit or after the time limit. SASE assumes that all the time is sorted by their timestamp, but in different types of RFID scenarios, this assumption is not established, it does not consider certain events in the RFID applications are to be effective only after a certain time period or within it.

Simple automata model cannot re-access the event after match an atomic event, if you need to consider the relationship about time between the various atomic events; you need to introduce a new data structure to hold this information, we can add the time limit on the basis of the finite state automata [1]. In order to derive the possible location of the RFID object and relationship of containment, we refer to timed automata (Time Automata, TA), to build its time-varying time-graph model for an original RFID data streams. Timed automata pro- posed by Alur and Dill [11], is a formal model for de- scribing, analyzing real-time system behavior. Timed Automata adds clock variables which can take real-valued based on finite state automata, using to depict the time of continuous change, it has a wide range of applicability. A paper [12] proposed a model based on timed automata using for Ad-hoc network intrusion detection. Another paper [13] proposed a modeling and verification of the Internet of Things services based on timed automata.

3. Definition

The background of this paper is the RFID smart library system [14], UHF readers installed in return book car can move among the bookshelves, by reading labels of bookshelves, layers and books they can know the location of different objects and infer constraints between objects. There are improvements for RFID data streams probable derivation method recently [15], however, this method cannot capture the relationships between objects, or provide the estimate of location.

Figure 1 is a schematic diagram of time functions in complex event, t_begin (e) shows the primitive event e occurs, t_end (e) shows the end of event e. interval (e1,e2) is time start from e1 and end by e2, distance (e1, e2) shows interval of e1 and e2.

Based on event specification described above, we now define RFID event and event streams for RTLS.

Definition 1. Events and Event Streams. Events reflect the current states of real-time system or states of the object changes, including the time of the incident and event type. An event type set $\varepsilon = \{E1, E2, \cdots, En\}$, an event is a tuple (E,t), for an RFID system, the event E = event

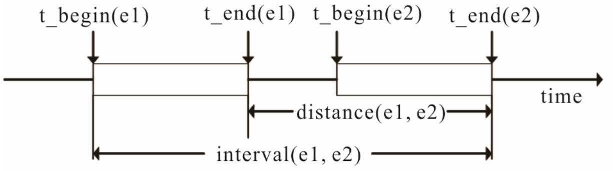

Figure 1. A diagram of time functions in complex event.

(tagID, readerID, timestamp), tagID is a label, readerID is a reader, the timestamp is the timestamp, in this $E \in \varepsilon$, *t* is the timing of the events. A steady stream of events forms the event streams (event streams), the event stream ES defined on ε is an event sequence sorted by time, ES = (E1, *t*1), (E2, *t*2), ··· (E*s*, *ts*), ···, where $ti < tj$ ($1 \le i < j \le s$).

Definition 2. Primitive Event (PE). Primitive events in RFID system is the underlying event, it is a data exchange between readers and tags, primitive events are inseparable. The formal definition of PE = {Ei| <ei, ti> ∈ Ei, by |Ei| = 1, the distance (PE) = 0, ti ≠ 0}.

Definition 3. Complex Event (CE). RFID complex events are senior events constructed by RFID primitive events. Formal definition for the CE = {Ei × Ej → E | <e,t> ∈ Ei × Ej, |E| ≠ 1, interval (ei, ej)≠0}.

Event stream is a simplify form continuous event data. When the event handler receives inflow event data, it applies real-time application logic on event data to expert knowledge of the behavior inherent in the event data. In this paper, with RFID-based real-time location applications, we propose a complex event processing. RTSL application background is defined as follows:

Definition 4. The physical world: $W(O, L, T)$. Said a geographic area, where O is the set of objects, L is the set of predefined position, T is the ordered discrete-time domain, L is the collection of the pre-defines logical location, or (*x*,*y*,*z*) coordinates generated by positioning system.

Definition 5. Relationship. At time *t*, W's state is embodied by the positional relationship set, a oi ∈ O appears in a li ∈ L, and there are constraints between the objects and object, the correspondence between the object and location.

Definition 6. Grid G. Known as the smallest unit of address space, denoted by G and G corresponds to a location li, is atomic.

Definition 7. Scenes S. We said address space unit corresponding to different levels called scenarios, denoted by S and S is constituted by the grid, that is G ⊂ S, contains relationship exists between scenes $S' \subset S$.

When the object's state changes (for example, into a region, to exit a certain area), or constraint relations between objects and objects are changed, the state of the scene will also change. In particular, the location of an object is unknown, if it does not appear in any of the predefined location, for example, passed between the two grids, or abnormal exit of the physical world.

The widespread deployment of the reader and other devices will generate a huge number of events, we need apply filtration and related association on these vast amounts of simple events, which in the semantic layer for end-user applications to detect more complex events, in order to achieve this purpose, we see these complex events as a continuous queries groups run on the simple events.

CEP language can make a detailed description of the event handling syntax and semantics, to make CEP engine easier to learn. CEP has designed a SQL extended language which supports continuous query to facilitate high-level application to submit their concern events. Here is a typical continuous event query:

```
EVENT StolenBookSTREAM
(bookshelf x, Borrow y, Exit z)
WHERE detect(x, y, z) {x. TagId = y. TagId and
                       y. TagId = z. TagId}
WINDOW 12 hours /* Library working hours */
RETURN x. TagId, x. ProductName, x. TimeStamp
```

The meaning of the above-mentioned continuous event query language is detection of library books without lawful to borrow to carry out from Museum. Detection of books on the theft is done by reading three specific RFID data streams readings: (1) taking books from the shelves; (2) didn't apply for borrowing procedures at librarian (or at the self-charging place); (3) book information is captured by Museum access system when getting out of Museum. EVENT clause DATA STREAM defined readings of the three data streams, WHERE clause specified assertion of events occurring based on multiple above-mentioned readings, in this case only if the same label appears in three RFID flows we confirm that detection event happens, we need restrict the primitive events streams through a special mechanism. Timed automata introduced by time constraints based on a finite state machine, to describe location changes of the RTLS.

4. RFID Complex Events Processing

4.1. The Definition of Timed Automata Theory

Timed Automata adds time constraints on finite automata, which can handle real-time system. Timed Automata uses a finite number of variables to represent the time, referred to as the time variable. It adds a constraint to annotate the state transition diagram, position migration in timed automata has clocked constraints, only when the clock constraint conditions are met, and the location migration can be carried out.

Definition 8. Timed Automata (TA). TA is a six-tuple $\langle S, s0, \Sigma, X, I, E \rangle$: Σ is a finite marker set; S is a finite set of states; $s0 \in S$ is the initial set of states; X is a finite set of clocks, each element can be expressed with the *x*, *y*, and z symbols; I said mapping, specify clock constraints in Φ(X) to S, that is I(s): $S \to \Phi(X)$; E is a convert set $E \subseteq \Sigma \times 2X \times \Phi(X) \times S$; a transition $(s, s', e, \lambda, \varphi)$ said an edge about event e from location s to location s', $e \in E$; collection $\lambda \subseteq X$ give the clock reset in the

$[\lambda \leftarrow 0]v$, this conversion to λ which is a subset of $x \in \lambda$, represents the clock interpretation: for each $x \in \lambda$, $([\lambda \leftarrow 0]v)(x) = 0$, and for each $x \in X/\lambda$, $([\lambda \leftarrow 0]v)(x) = v(x)$; Φ is an clock constraints on X specify when state transition occurs, to a clock variable finite set X, an X clock interpretation is a mapping v: X→T, it allocates each clock a time value. Semantic of timed automata is defined by a conversion system STA, which associated with it. A state of STA is a tuple (s,v), s is a TA position, v is a clock interpretation of X, and v satisfy the mapping I(s). The set of all states of TA is expressed QTA. If an initial position of the TA and v(x) = 0, then the state (v,x) is an initial state.

There are two types of conversion in STA:

1. Time migration. For a state (s,v) and a real-valued time increment $\delta \geq 0$, if $0 \leq \delta' \leq \delta$, $v + \delta'$ all met the mapping I(s).

2. Location migration. To (s,v) and $(s, s', e, \lambda, \varphi)$, if v satisfy φ.

Time migration, said the passage of time in a node, the location migration said the position conversion between the nodes if the constraint conditions are met. Migration is divided into strong and weak, strong migration on behalf of detection events and the next event is a partial order, expressed with a solid line; weak migration on behalf of detection events and the next event is ordering relation, expressed with the dashed line.

4.2. The Derivation of the Event Streams Constraints

When RFID Reader of RTLS reads the temporal and spatial information data, it requests the data values can reflect the current state of the real world; we need a consistent mechanism to ensure the time consistency of spatial data.

In order to derive the probable location of the object and probably contains relationship, we refer Timed Automata [10] to build time-varying time graph model for the original RFID data streams. G = (V, E, T) is the coding of the objects in the physical world (set of edges to be updated), contains the location of the report and possible constraints, at the same time contains statistical and historical synergy between objects. The node set V represents all of the physical world objects with RFID tags, objects exist in multiple levels of the scene: the library stacks, shelves, books, they are arranged according to the geographical space, each scene corresponding to a layer, known objects remain in the corresponding scene on behalf of a known location, expressed by a color, objects whose location unknown did not stain. According to the readings on the flow we update the color of the node. If in a given moment an object is not read by the reader, the corresponding node color is empty, the value of the node is a tuple (R_t, timestamp), R_t is reading set. These uncolored nodes keep the memory of recent color and the corresponding observing time.

The edge set E packet the possible constraint relations between objects, the edge oi → oj, express oj contains oi, (for example, bookcase containing layer, layer contains books) ,we allows each node to have multiple input and output side on behalf of the shelves can contain multiple books, one book could appear in many shelves. We allow the merger of the colored and uncolored nodes. An edge cannot link two nodes of different colors. That is not allowed objects which stay in two different locations establish constraint relationship.

To support the derivation, this time figure also comes with statistics information, each edge maintains a bit vector recording the last arrangement information of two objects. When two points linked by an edge assigned the same color, we set the corresponding bit value. Each node records statistical information of the parent node, or to use probabilistic methods [14] to deduce the number of inconsistent observations so far, in all input side, to allow only a final confirmation line, expressed by the solid line.

Time is divided into time slices, each time slice corresponds to a moment, in each moment algorithm uses time graph G before that and generates a new graph G^* with the readings set R^k, we allocate clock for each conversion, using automatic machine to move from one state to another state, if the time for the introduction of the event to meet the time constraints, we will assign time value to each conversion. After each migration (transformation), the time graph will reset the time, that is, reset the clock. The system should be able to control the redistribution of the clock in the time diagram. RFID data streams timing diagram is shown in **Figure 2**.

Update algorithm of data stream timing diagram meets the request—"one pass", that is, when each reading arrives we update map by increment, and to ensure that at some point the graph G^* consistent with the readings from all readers, the update process of graph is divided into three steps:

4.2.1. The Initial State S1: Update and Color Nodes

If the new object was first observed, then create a new node in the graph; for each observed object, using the observed color of the reader to color the corresponding node, objects not observed cannot be updated. At the same time update statistics of the edge which at least is a node colored.

4.2.2. Update the Status S2: Update the Edge

If the two nodes of the adjacent layers have the same color, if there is no edge, then add an edge between the two nodes, thus enumerate all of the constraints (for ex-

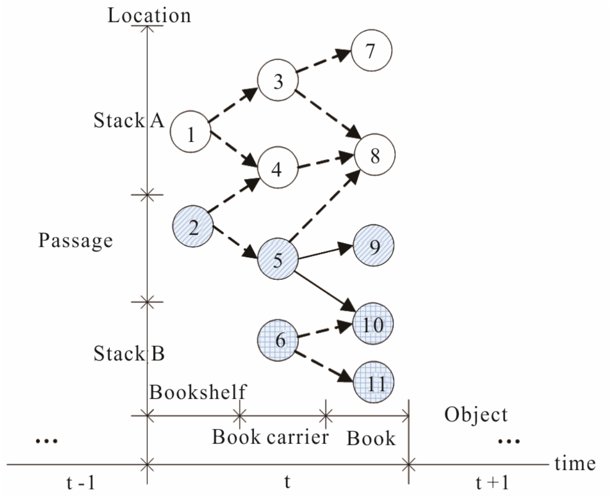

Figure 2. Timing diagram model of event streams for RTSL.

ample, slash-filled books can be included in the slash-filled book cart), moment t ②→⑤ in **Figure 2** expressed an edge created when the book car is near the shelf.

4.2.3. Optimization of State S3: Pruning on the Graph

If the nodes of an edge are assigned different colors that objects in the same location previously are in different positions according to the latest arrival of the readings, this edge will be removed from **Figure 2**. The accuracy of the improvements of timing diagram mode will help cut unnecessary edges.

Complexity analysis of algorithm: in time t, reading set is R_t, cost for updating the node is less than $|R_t|$; each reader reading set is R_t^k, cost of adding edge for adjacent node in the same location is less than $\left|R_t^k/2\right|*2$, updating cost of set of edges G to be updated is $2*\left|\pi_{R1,R2,\ldots,Rk}G\right|$.

5. Simulation and Performance Evaluation

In order to test the performance of the GEEP, RFID equipment is deployed in a real library environment, including readers (902 - 928 MHz), labels, number of samples is 1000 books, 20 bookshelves, shelves with tags, each bookshelf with 8 layer tags, the books are contained in a layer. The test range is 100 M^2.

The data acquisition server configuration: Pentium 4 3.0 G/1 G/100 G, event handling server configuration Pentium Core Duo, 1.8 G/2 G/100 G.

5.1. Experiment 1: Accuracy Test

Contrasting with event detection engine achieved by the Microsoft RFID middleware BizTalk, we record reading which randomly assigned to books in the process of moving and contrast that with the actual location of the books to test the infer accuracy of GEEP.

The experimental results were shown in **Figure 3**. Experiment was repeated 10 times, the framework of RFID data stream can be greater than 95% accuracy of correctly to infer the object's location and inclusion relations, higher than the accuracy of inference based on BizTalk, prove the effectiveness of time graph derivation and streaming continuous query technology.

5.2. Experiment 2: Performance of Response Test

To simulate automatic inventory scenarios, we deploy some fixed reader in proper locations to detect books unreasonable move events. The specific process is to remove a book from a bookcase on a layer, move to the other shelves and insert the corresponding layer, then identify if the moving event is reasonable.

Figure 4 shows that the GEEP event detection engine can detect events less than 0.2 seconds delay (mean value is 0.1422 seconds). That is approximate real-time response. Experiment 2 demonstrates that the average measure of the GEEP event detection engine delay can reach the millisecond level, can be applied to the occasions count in millisecond.

5.3. Experiment 3: Processing Speed Test

On the basis of the actual RFID equipment acquisition tag data, we simulated a large number of load, when the simulation load reach more than 50,000 objects, we can still stay on the line response to events (less than 1 second), such as shown in **Table 1**. Compression technology proposed in this paper effectively reduce the amount of calculation and the framework has good scalability and can handle 100,000 messages per second in the dual-core CPU environment, we can infer in multi-CPU or distri-

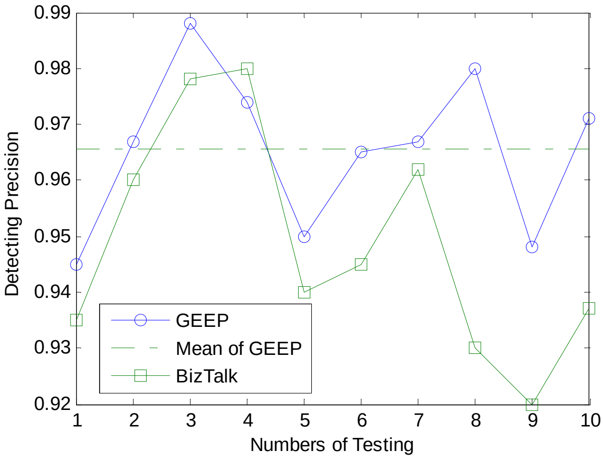

Figure 3. Contrast for inferring accuracy.

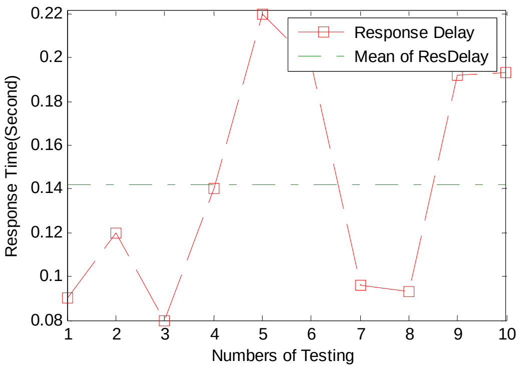

Figure 4. GEEP response time.

Table 1. Cost for updating graph and event inference.

Number of load	Cost for updating graph(s)	Cost for event inference(s)
10,000	0.00 134	0.05 132
20,000	0.00 312	0.11 343
30,000	0.00 523	0.17 863
40,000	0.01 122	0.21 331
50,000	0.01 945	0.62 811

buted environment. The performance will reach one million messages per second.

6. Conclusion

In this paper, we describe the event stream constraint graph model based on Timed Automata; it can capture the probable object location and constraints relationship. Then evaluate its performance, performance evaluation proves the proposed method can effectively detect and deal with complex events, can be widely used in air transport regulation, financial risk management, network intrusion detection, time positioning systems and other areas. Timestamp of out-of-order phenomenon is very common in RFID applications messy chronological order may result in errors of complex event detection [1]. Time automata theory will be used in complex event detection in RFID event stream out of order in the future.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 60803001, the Jiangsu Province Graduate Creative Foundation under Grant No. AD20255. The part of this work was presented in IEEE MINES2012 conference.

REFERENCES

[1] Y. Gu, G. Yu and T. C. Zhang, "RFID Complex Event Processing Techniques," *Journal of Computer Science and Frontiers*, Vol. 1, No. 3, 2007, pp. 255-267.

[2] C. Guo, K. Li, Y. Y. Wang, S. H. Liu and H. A. Wang, "A Location Index for Range Query in Real-Time Locating System," *Journal of Computer Research and Development*, Vol. 48, No. 10, 2011, pp. 1908-1917.

[3] F. S. Wang, S. R. Liu and P. Y. Liu, "Complex RFID Event Processing," *The VLDB Journal*, Vol. 18, 2009, pp. 913-931. doi:10.1007/s00778-009-0139-0

[4] W. Yao, C.-H. Chu and Z. Li, "Leveraging Complex Event Processing for Smart Hospitals Using," *Journal of Network and Computer Applications*, Vol. 34, No. 3, 2011, pp. 799-810. doi:10.1016/j.jnca.2010.04.020

[5] D. C. Luckham, "The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems," Addison-Wesley Longman Publishing Co., Inc., Boston, 2001.

[6] F. Fabret, H. Jaconsen, P. Lirbeta, K. Ross and D. Shasha, "Filtering Algorithm and Implementation for Very Fast Publish/Subscribe System," *Sigmod*, Vol. 30, No. 2, 2001, pp. 115-126. doi:10.1145/375663.375677

[7] E. Wu, Y. Diao and S. Rizvi, "High-Performance Complex Event Processing over Streams," *SIGMOD*, Chicago, 27-29 June 2006, pp. 407-418. doi:10.1145/1142473.1142520

[8] M. Altinel and M. Franklin, "Efficient Filtering of XML Documents for Selective Dissemination of Information," *Proceedings of the* 26th *International Conference on VLDB*, Cairo, 10-14 September 2000, pp. 53-64.

[9] SAX Project Organization, "SAX: Simple API for XML," 2004. http://www.saxproject.org/

[10] Y. Diao, M. Altinel, M. Franklin, H. Zhang and P. Fischer, "Path Sharing and Predicate Evaluation for High-Performance XML Filtering," *ACM Transactions on Database Systems*, Vol. 28, No. 4, 2003, pp. 467-516. doi:10.1145/958942.958947

[11] R. Alur, L. Fix and T. A. Henzinger, "Event-Clock Automata: A Determinizable Class of Timed Automata [C]," *Proceeding of the* 6th *Annual Conference on Computer aided Veification* (*CAV*). *Lecture notes in computer science* 818, Springer-Verlag, London, 1994, pp. 1-13.

[12] Y. Ping, L. Ning, W. Yue, "Intrusion Detection Based Timed Automata for Ad Hoc Networks," *Journal of Electronics & Information Technology*, Vol. 31, No. 10, 2009, pp. 2310-2315.

[13] L.-X. Li and Z. Jin, "Modeling and Verifying Services of Internet of Things Based on Timed Automata," *Chinese Journal of Computers*, Vol. 34, No. 8, 2011, pp. 1366-1377.

[14] A. Repanovici, "Smart Library: RFID Implementation in Libraries [C]," *Proceedings of the* 8th *WSEAS International Conference on Artificial Intelligence*, *Knowledge Engineering and Data Bases*, Cambridge, 21-23 February 2009, pp. 523-526.

[15] Y. M. Nie, Z. H. Li and Q. Chen, "Enhanced Probabilistic Interpretation over Uncertain Radio Frequency Identification Data Streams," *Journal of Xi'an Jiaotong University*, Vol. 45, No. 12, 2011, pp. 45-52.