

The Research on CEP Based on Query Rewriting for CPS

Kening Cao, Renfa Li, and Fengjuan Wang

Abstract—Now we are entering the era of "Big Data", and the important data source of which is Cyber Physical System (CPS) data, those data can hardly be processed by present data processing technology. Complex Event Processing (CEP) is the key part of middleware of CPS, which can process the big data, and context-awareness is an important feature of CEP engine. There are some challenges in context-aware complex event processing for CPS, such as heterogeneous, massive data, distributed event processing, uncertainly; therefore, we propose a distributed context-aware event processing method, based on context representation proposed fuzzy ontology and a distributed multi-level context reasoning method to support uncertainty and linguistic variables in event queries. Based on fuzzy ontology query and similarity based distributed reasoning, complex event query plans are generated and context-aware queries are rewritten into context independent sub-queries. Data window is partitioned according to different event patterns and context. The sub-queries are optimized and executed parallel based on data partition. The experiments show that this method can support fuzzy context in CEP and have acceptable performance and scalability.

Index Terms—CPS, complex event processing, context-aware, query rewriting

I. INTRODUCTION

With the rapid development of information, we are entering the era of "Big Data", Big data has the characteristics of huge data, massive of data types, low density of value and real-time requirements. The important data source of big data is Cyber Physical System (CPS) application. Which has established more than 30 million sensor network that consist of sensors, and the speed rate of growth is more than 30% per year [1]. So enormous events are produced by various kinds of devices at high speed In a large scale CPS application, there might be various kinds of devices that generate events such as sensors which constitute the wireless sensor network, RFID tags and readers, GPS, etc. In real CPS application, we pay more attention to complex information such as business logic and rule. Complex Event Processing (CEP) [2] is used to process huge primitive events and get valuable information from them. So CEP engine has a key role in processing of CPS data. Context plays the same role in event processing that it plays in real life. A particular event can be processed differently depending on the context in which it occurs. Context representation and reasoning has been studied widely in context aware computing and

ubiquitous computing. But recently few papers can be found about context aware complex event processing. Some papers provide some idea and framework but no detailed information about how to implement. Primitive events are generated from RFID reader or sensor data streams. It is widely known these data streams include noise, so the interpretation has more than one possible candidate. The uncertainty of event is usually treated with probabilities. Context can also be uncertain. Therefore it is important to develop complex event processing engine that be able to deal with the uncertainty in events and context.

Now there are some research on CEP and context-awareness, Most of the CEP methods in active databases and RFID applications use fixed data structure. Yuanping et al. used tree-based CEP method and optimized the algorithm by event grouping [3]. Fusheng et al. used directed graph to process complex event in RFID stream [4]. Xingyi et al. used TPN (Timed Petri-Net) to detect complex event from RFID stream [5]. Recently, there is still much work on Petri-Net based CEP since uncertain event is important in many applications. Eugene proposed a query plan based complex event detection method SASE (Stream-based And Shared Event processing) [6], Haopeng improved SASE model to support imprecise timestamps when processing complex events in stream [7]. However, these researches mostly adopt traditional centralized CEP architecture, which don't support distributed application, and the system is hardly robust and scalable. On the other hand, some CPS applications are naturally distributed and we need to detect complex events from the distributed system. Distributed CEP architecture over CORBA and a plan based distributed CEP algorithm have been proposed in [8], it uses parallel method to improve the performance of distributed complex event processing, but it cannot support uncertainty data.

Context modeling is a fundamental step for every adaptation application and the development of context-aware systems. Many models have been developed including key-value model, object-oriented model and ontology model. Ontology is considered the best model for context representation [9]. Almeida et al. presented an ontology that models the ambiguity in intelligent environments based on fuzzy logic [10]. Singh et al. proposed a Fuzzy Integrated Ontology Model which aims to integrate fuzzy logic in design structure of ontology [11]. Zhang et al. proposed a distributed fuzzy reasoning Petri net model towards the decomposition of distributed fuzzy reasoning [12]. In open heterogeneous environment, context-awareness is one of the key requirements.

The main challenge in the context-aware system is to make appropriate decisions by promptly taking into account the user's context, which are the data used to characterize the user's current situation. The task of manipulating the context

Manuscript received May 9, 2013; revised July 21, 2013. This project is sponsored by "complex event processing in large scale internet of things (K120326-11)" project of Changsha technological plan and "Context-aware active complex event processing in large scale internet of things (13JJ3046)" of natural science foundation of Hunan province.

The authors are with College of Information Science and Engineering, Hunan University, Changsha, 410082, Hunan, China. (e-mail: 9462478@qq.com and wang_fengjuan31@163.com)

data in an intelligent way is a crucial contemporary research, which is often referred to as context reasoning. ChunKyungLee *et al.*, propose similarity-based context reasoning to define the similarity between the context models [13]. This does not require the original context information to reduce the complexity of reasoning process. In addition, the correlation of the context models was taken into account using Mahalanobis distance for similarity-based distributed reasoning, the trust of reasoning outcomes was enhanced by finding an optimal context model and avoiding the ambiguity between two context models.

Recently query rewriting has been studied in the new area of ontology reasoning and event processing. Ella *et al.*, presented a formal model for pattern rewriting and demonstrated its usage in a comprehensive set of rewriting techniques for complex pattern types, taking various semantic interpretations into account [14]. Nicholas et al. proposed a distributed complex event processing method based on query rewriting [15]. In their work, event patterns are specified in a high-level query language and, before being translated into event automata, are rewritten in a more efficient form. Automata are then distributed across a cluster of machines for detection scalability. Eric et al. introduced filter-based rewrite rules similar to predicate push-down in query optimization of Schema Matching Processes [16]. Modeling tool and recommendation systems for rewriting matching processes are also proposed in their work. Domenico et al. provided a query rewriting method for inconsistent DL-Lite ontologies [17]. In their work, some variants of inconsistency-tolerant semantics are reduced to standard evaluation of a FOL/SQL query over a database through query rewriting. But recently there are few papers about context-aware CEP with query rewriting.

There are some challenges in complex event processing for big data in CPS, such as complexity of the event type and system structure, uncertainty, and mass properties, in order to resolve the problems; we propose a high performance distributed context-aware complex event processing method (DCACEP) for CPS.

II. EVENT AND CONTEXT MODEL

Definition 1. Complex event query language: A pattern query addresses a sequence of events that occur in temporal order and are further correlated based on their attribute values.

The overall structure of our research language is:

```
PATTERN <pattern structure>
[WHERE <pattern matching condition>]
[WITHIN <time window>]
[RETURN <output specification>]
```

Example 1.query Q1:

```
EVENT SEQ(A × 1, B × 2, C × 3, D × 4)
WHERE [attr1, attr2] x1.attr3 = 'x1.attr4 < x4.attr4
skip_till_any_match (A, B, C, D)
Within T
Return A, B, D
```

Where A, B, C, D represent four events of the specified types occurring in temporal order. The WHERE clause further specifies constraints on these events: 1) attr1, attr2 are the common attributes, and attr1=attr2; 2) the attr3 of A is a

input value; 3) the attr4 of A is less than the attr4 of D.

Definition 2. Primitive event: A primitive event in event cloud means an atomic occurrence of interest in time. A primitive event is represented as $\langle A, T \rangle$ where A is the set of attributes and T is the timestamp that the event occurs.

Definition 3. Complex event: Complex event is a combination of primitive events or complex events by some rule. A complex event is represented as $\langle E, R, T_s \rangle$ where E represents the elements that compose the complex event, R represents the rule of the combination, T_s represents the time span of the complex event. The main complex event patterns in our work include "All", "Any", "Count", "Seq", etc.

Definition 4. Context model: A context is a named specification of conditions that groups event instances so that they can be processed in a related way. It assigns each event instance to one or more context partitions. A context may have one or more context dimensions and can give rise to one or more context partitions.

Example 3. Event context types: Context C_1 : Within 2 km from the motel M_1 . Context C_2 : Within 10 km from accident A_1 . Context C_3 : traffic status of highway is traffic slow (Traffic in a certain highway has several status values: traffic flowing, traffic slow, and traffic stationary). C_1 is an entity distance context. C_2 is an event distance context and C_3 is a state-oriented context.

III. DISTRIBUTED CONTEXT-AWARE EVENT PROCESSING

A. Fuzzy Ontology Based Context Representation

The context representation in our work is based on the fuzzy ontology framework of [14] and optimized for event processing.

Definition 5. Fuzzy ontology: A fuzzy ontology O in a particular domain Δ is $O\Delta = (C, R, P, I, A)$ where C is a set of fuzzy concepts, P is a set of fuzzy properties of concepts, I is a set of objects, R is a set of fuzzy roles which are the relations between two objects and A is a set of axioms that expressed in a logical language.

Definition 6. Fuzzy concept: A fuzzy concept C is defined as $C = \{a_1v_1, a_2v_2, \dots, a_nv_n\}$ where a_i is an object, v_i is the membership degree of object i in concept C. The degree of object a belongs to a fuzzy concept C is give by a fuzzy membership function $\mu_C: A \rightarrow [0, 1]$ where A is the set of objects.

Definition 7. Concept subsumption: For two fuzzy concepts $X = \{a_1w_1, a_2w_2, \dots, a_nw_n\}$ and $Y = \{a_1v_1, a_2v_2, \dots, a_nv_n\}$, a_i is an object, w_i and v_i is the membership degree of object. If $\forall a_iw_i \in X, a_iv_i \in Y, v_i \geq w_i$ then X is subsumed by Y which is denoted by $X \in Y$.

Definition 8. Fuzzy role: A fuzzy role R is a fuzzy set of binary relations between two objects in the domain. It is interpreted as a set of pairs of objects from the domain denoted by $R = \{ \langle a_1, b_1 \rangle w_1, \langle a_2, b_2 \rangle w_2, \dots, \langle a_n, b_n \rangle w_n \}$ where a_i and b_i are two objects, w_i is the degree of the strength of the relation which is given by a fuzzy membership function $\mu_R: A \times B \rightarrow [0, 1]$, where A and B are set of objects. The set of object A is named the domain of the role while the set of objects B is named the range of the role.

Definition 9. Fuzzy role subsumption: For two fuzzy roles $S = \{ \langle a_1, b_1 \rangle w_1, \langle a_2, b_2 \rangle w_2, \dots, \langle a_n, b_n \rangle w_n \}$ and $T = \{ \langle a_1, d_1 \rangle v_1, \langle a_2, d_2 \rangle v_2, \dots, \langle a_n, d_n \rangle v_n \}$, if $\langle a_i, b_i \rangle w_i \in S, \langle a_i, b_i \rangle v_i \in T, v_i \geq w_i$ then S is subsumed by T denoted as $S \in T$.

Definition 10. Fuzzy property: A fuzzy property P is defined as $P = R \cdot C$ where R is a fuzzy role, C is a fuzzy concept which is the range of the fuzzy role.

Definition 11. Fuzzy property subsumption: For two fuzzy property $P_1 = \{(\langle a, c \rangle, c) \vee i | (\langle a, c \rangle, c) w_1 i \in S, cy_1 i \in C\}$ and $P_2 = \{(\langle a, c \rangle, c) \vee i | (\langle a, c \rangle, c) w_2 i \in S, cy_2 i \in D\}$, if $(\langle a, c \rangle, c), (\langle a, c \rangle, c) \vee i \in P_1, (\langle a, c \rangle, c) \vee i \in P_2, v_2 i \geq v_1 i$, then p_1 is said to be subsumed by P_2 , denoted by $P_1 \in P_2$.

Definition 12. Fuzzy property with linguistic variable: A fuzzy property can be represented by $\langle V, M \rangle$ where V is a set of linguistic variables and M is the membership function.

Example 4. fuzzy ontology example: 'red car' is a fuzzy concept and it is subsumes another fuzzy concept 'car'. 'Bob extremely like sport car' is a relation of a fuzzy role and the degree of the strength of this relation is very high (extremely). 'supervise.master-student' is a fuzzy property and it is subsumed by another fuzzy property 'teach.student'. Fuzzy property 'drive.speed' can be set to $\langle \{Slow, Medium, Fast\}, \{S, M, F\} \rangle$ and S, M, F are depicted in (1)

$$S = \int_0^{100} \left(\frac{100-x}{100} \right) / x$$

$$M = \int_0^{100} \left(\frac{x}{100} \right) / x + \int_{100}^{200} \left(\frac{200-x}{100} \right) / x \quad (1)$$

$$F = \int_{100}^{200} \left(\frac{x-100}{100} \right) / x$$

Definition 13. Fuzzy context: Fuzzy context denoted by FC is defined as a triple $FC = \langle N_c, N_o, N_p \rangle$ where N_c is a set of fuzzy concepts, N_o is a set of objects and N_p is a set of fuzzy properties.

A part of traffic domain fuzzy ontology is shown in Fig. 1. The fuzzy ontology is represented based on Fuzzy OWL and the basic ontology reasoning component is created based on FuzzyDL.

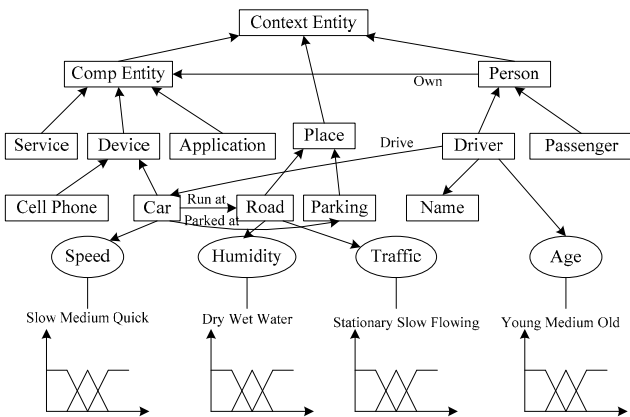


Fig. 1. A simplified traffic domain fuzzy ontology

B. Distributed Multi-Level Context Reasoning Method

In the large-scale agent system, the logic-based reasoning methods become complex and difficult to gain an overall picture of the context model. In order to support big data, we propose a distributed similarity based reasoning method based on the multi-level context model shown in Fig. 2.

In this model, the level 1 context state is generated from the fuzzy ontology instances and the later are created from

basic CEP engine. Ontology instances are distributed since there may be many event processing agents. A level $i+1$ context state can be deduced from lower level contexts.

- Higher level context reasoning

The uncertainty reasoning process that deduces higher level context state from lower level context is designed based on fuzzy D-S evidence theory [18].

Definition 14. expectation certainty: expectation certainty EC (B) can be calculated by

$$EC(B) = \sum_i m(A_i) \cdot \inf(A_i \Rightarrow B) \quad (2)$$

where m is the mass function. When A_i and B both are not fuzzy sets, EC (B) will degenerate into the belief and plausibility degree of the standard D-S theory.

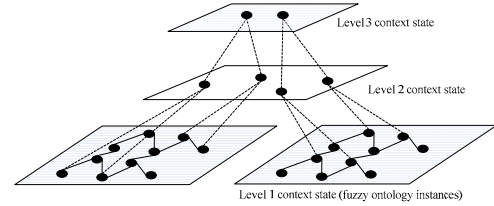


Fig. 2. Multi-level context model

Definition 15. Inclusive measurement: inclusive measurement $I (A \subset B)$ is the degree to which set A is included in set B. It can be calculated by

$$I(A \subset B) = \frac{\min_x \{1, 1 + \mu_B(x) - \mu_A(x)\}}{\max_x (\mu_A(x))} \quad (3)$$

Then the belief function can be calculated by

$$Bel(B) = \sum_A I(A \subset B) m(A) \quad (4)$$

Definition 16. Extended Dempster combinational rule: The Dempster combinational rule can be extended as follows,

$$m_1 \oplus m_2(C) = \frac{\sum_{A \cap B = C} \max_{x_i} \mu_{A \cap B}(X_i) m_1(A) m_2(B)}{1 - \sum_{A, B} (1 - \max_{x_i} \mu_{A \cap B}(X_i)) m_1(A) m_2(B)} \quad (5)$$

Based on the extended Dempster combination rule, low level context (evidence) can be combined to deduce high level context (conclusion). A collision factor is used to resolve the Zadeh's paradox. We also use an evidence selection strategy to optimize the performance.

- Similarity based distributed context reasoning

In the similarity based distributed context reasoning process, high level context among different CEP engines is merged using D-S evidence theory. During context updating process, new context is merged into existing context based on the fuzzy sets similarity.

Definition 17. Fuzzy sets similarity: suppose $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ to be the frame of discernment, and the fuzzy sets \tilde{A} and \tilde{C} are the two fuzzy subsets, then the similarity between fuzzy \tilde{A} and \tilde{C} is defined as:

$$\omega(\tilde{C}, \tilde{A}) = 1 - \frac{1}{|\Theta|} \sum_i |\mu_{\tilde{C}}(\theta_i) - \mu_{\tilde{A}}(\theta_i)| \quad (6)$$

Select Evidence method is executing D-S evidence select theory. Processing low-level context step by step, and cluster to generate high level context based on similarity of definition 17. ClusterEvidence method use dynamic plan method to guarantee the performance of local context update. A negotiation protocol is used to control the transportation of context among distributed CEP engines. The protocol try to make the overall data flow of the network to be minimum and the load of every distributed node to be balanced. When context is changed, new high level context is generated

Using this reasoning process we can get high performance distributed reasoning. There is no global control CEP engine which means the algorithm is totally distributed. Traditional distributed reasoning process applies the partitioning and merging whenever the context is updated. Therefore, the reasoning process becomes complex when a large number of agents are involved. In our method, new context can be merged into the most similar existing context and the existing context in different branch will not be affected.

Algorithm 2. distributed query rewriting method

Input: EC is the primitive event stream, Q is the query clause

Output: EI is the event instances satisfies the query

Method:

```

subQueries ← GlobalResolve(Q, globalContext)
isControlNode ← GetControlNode(Q)
partitions ← Partition(EC, nodes, globalContext)
for i = 1 to nodesNumber do
    results[i] ← subCEP(partitions[i], subQueries[i],
        isControlNode)
end
EI ← MergeResult(results, globalContext)
Return EI
SubCEP(SC, SQ, isControlNode):
Input: SC is sub_event stream, SQ is CEP the query
clause, is_control_node is mean whether need
sub_control node
Output: SEI is CEP patterns sub_results
Method:
SEI ← Φ, event_filter ← Φ, temp_result ← Φ
eventPattern ← AnalysisQuery(SQ)
eventContext ← AnalysisContext(SQ)
for each sec eventContext do
    ResolveContextParameters(sec)
    sc ← DecomposeContext(sec)
    eventFilter ← eventFilter ResolveEventFilter(sec)
end
dataWindow ← Filter(eventFilter, SC)
eventPartitions ← PartitionEvent(dataWindow,
eventContext, eventPattern)
subQueries ← GenerateSubQuery(eventPartitions,
eventContext, eventPattern)
i ← 1
for each q sub_queries do
    result[i] ← executeParallel(q)
    i ← i + 1
end
    
```

```

SEI ← MergeResult(result)
if(isControlNode)
    controlNode ← getControlNode()
    if(controlNode == localNode)
        tempResult ← ReceiveTempResult()
        SEI ← MergeTempResult(TempResult, SEI)
    else
        SendTempResult(SEI)
    end if
end if
Return SEI
    
```

C. Distributed Query Rewriting Method

The query rewriting method in our work is shown in algorithm 2. Complex event query clause is analyzed by syntax parser and syntax tree is created. Event patterns and context are included in the syntax tree. Both pattern and context can be composite.

The algorithm 2 can be partitioned into 4 steps which are query analysis, context resolving and event stream filtering, data partitioning, query plan generating and executing. Some context contains parameters and we need to resolve the parameters first. The parameters can be resolved in three ways: event database, ontology and sub-queries.

There are 3 types of event partition methods. The first method is partition by spatial context. In this method, the space is partitioned into N areas first. The event sequences within an area are partitioned into that area. All the event sequences across more than one area are portioned into the number N+1 area. This partition method can make sure there is no overlap among areas. The second is based on segmentation context. The third is based on temporal context. The time window of events can be partitioned into M parts in this method.

In query plan generating and executing step, a sub-query is generated for each partition. The sub-queries are then executed parallel and finally the result of all sub-queries is merged into a final result.

In the distributed CEP system, sometimes the local CEP engine needs data from other nodes. We transport the temporary result but not the original data. In order to save the network bandwidth and avoid overload of the global node, instead of sending all the temporary result to the global node, the algorithm select a proper local node as the control node (the get Control Node() method) and send the temporary result to that node.

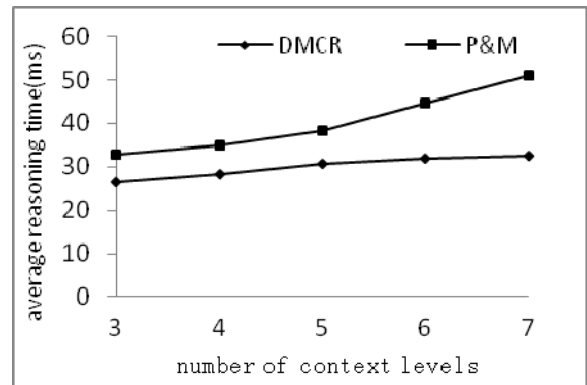


Fig. 3. Event partition by spatial context

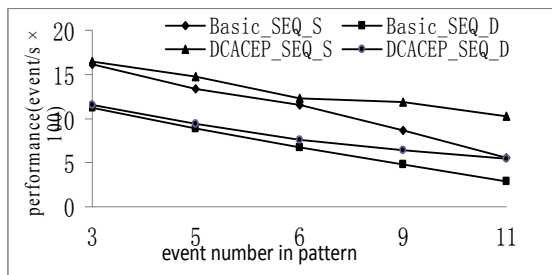


Fig. 4. Performance comparison of basic method and DCACEP on dense data set and sparse data set

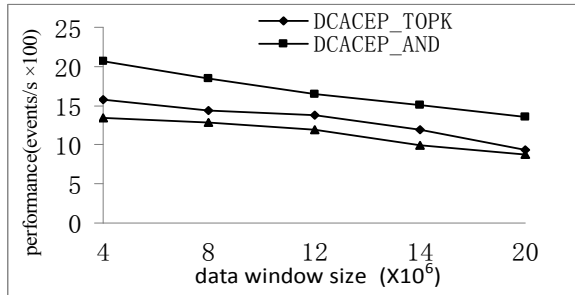


Fig. 5. Scalability of DCACEP on sparse data set

IV. EXPERIMENTAL EVALUATION

We developed a simulator which can generate primitive events for different kind of sensors and RFID tags according to the device configuration. The objects in the simulator move according to the policy configuration. The CEP engine can read primitive events from the simulator and then detect event patterns according to user's query. In order to support fuzzy context, we created a simplified fuzzy ontology for internet of vehicles for a small virtual city. We used two Lenovo servers with 4GB memory and two Intel Core 2 Duo processors in our experiment.

The performance of DCACEP is related to event selecting factor which is defined as $S_f = N_p/N_w \times 100\%$ where N_p is the number of instances for the complex pattern and N_w is the size of the data window. In our experiments we use dense data set ($S_f > 10\%$) and sparse data set ($S_f < 0.1\%$).

We first evaluated the performance of the distributed multi-level context reasoning method (DMCR). We implemented Amir's "partitioning and merging" method [24] in our system (we call it P&M) and compare it with our reasoning method. The result is shown in Fig. 3 We can see the performance of DMCR is better than P&M and when the number of context level increase, the running time of DMCR does not increase as much as that of P&M. The reason is that DMCR use the similarity based method to avoid recalculating for many high level context nodes.

In the second experiment, we compared the performance between dense data set and sparse data set which is shown in Fig.4. In the figure '_D' means dense data set and '_S' means sparse data set. We can see the performance of sparse data set is better than that of dense data set. The reason is that the instances of NFA or tree for dense data set are large. The performance of both DCACEP and basic method decreases when the event number increase for sparse data set. But the performance of DCACEP decreases slowly because of the query optimization.

In the last experiment we studied the scalability of

DCACEP on sparse data set which is shown in Fig.5. We selected 'SEQ', 'AND' and 'TOP-K' patterns. As we can see, the performance does not decrease much when the data window becomes larger to a certain extent. DCACEP has good scalability because it uses a optimized totally distributed similarity base context reasoning method and the query rewriting method is also optimized to make full use of multiple servers and processors.

From the experiments we can conclude that DCACEP can support context in distributed complex event processing and the performance and scalability is acceptable for large scale application

REFERENCES

- [1] J. Manyika, M. Chui, and B. Brown. Big data: The next frontier for innovation, competition, and productivity McKinsey. [Online]. Available: http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation
- [2] D. Luckham, "The power of events: an introduction to complex event processing in distributed enterprise systems," *Lecture Notes in Computer Science*, 2008, pp. 3.
- [3] Y. Li, J. Wang, L. Feng, and W. Xue, "Accelerating Sequence Event Detection through Condensed Composition," in *Proc. the 5th International Conference on Ubiquitous Information Technologies & Applications*, Sanya, China, Dec. 2010.
- [4] F. H. Wang, S. R. Liu, P. Liu, and Y. J. Bai, "Bridging physical and virtual worlds: Complex event processing for RFID data streams," in *Proc. the 10th International Conference on Extending Database Technology (EDBT'2006)*, 2006, pp. 588-607.
- [5] X. G. Jin, X. D. Lee, K. Ning, and B. P. Yan, "Efficient Complex Event Processing over RFID Data Stream," in *Proc. the Seventh IEEE/ACIS International Conference*, 2008, pp.75-81.
- [6] E. Wu, Y. Diao, and S. Rizvi, "High-performance complex event processing over streams," in *Proc. the 2006 ACM SIGMOD International Conference on Management of Data*, 2006, pp.407-418.
- [7] H. Zhang, Y. Diao, and I. Neil. Recognizing Patterns in Streams with Imprecise Timestamps, *PVLDB*, vol. 3, no. 1, pp. 244-255. Sept. 2010.
- [8] Y. H. Wang and S. H. Yang, "Plan based distributed complex event processing for RFID application" in *Proc. Computational Intelligence and Software Engineering*, 2009, pp.1-4.
- [9] M. Miraoui, C. Tadj, and C. B. Amar, "Context modeling and context-aware service adaptation for pervasive computing systems," *International Journal of Computer and Information Science and Engineering (IJCISE)*, vol. 2, no. 3, pp.148-157. May 2008.
- [10] N. Daei, H. M. Shirazi, R. Askari, and M. Ghanavati, "Service robot navigation based on Q-learning and fuzzy logic," *International Journal of Robots, Education and Art*, vol. 1, no. 2, pp. 1- 9, 2011.
- [11] A. Singh, D. Juneja, and A.K. Sharma, "A fuzzy integrated ontology model to manage uncertainty in semantic web: The FIOM," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 3, no. 3, pp. 1057 - 1062. 2011.
- [12] H. Zhang and K. Chen, "Building social relationship ontology model based on fuzzy set," *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 15, pp. 459- 466, 2012.
- [13] A. Padovitz, S. Wai Loke, and B. Arkady. Zaslavsky, "Multiple-agent perspectives in reasoning about situations for context-aware pervasive computing systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 38, no. 4, pp. 729-742 .2008.
- [14] E. Rabinovich, O. Etzion, and A. Gal, "Pattern rewriting framework for event processing optimization," in *Proc. the 5th ACM International Conference on Distributed Event-Based Systems*, New York, 2011, pp.101-112.
- [15] P. Nicholas, S. Møller, M. Migliavacca, and P. Pietzuch, "Distributed complex event processing with query rewriting," in *Proc. the Third ACM International Conference on Distributed EventBased Systems*. 2009, pp.147-156.
- [16] E. Peukert, H. Berthold, and E. Rahm, "Rewrite techniques for performance optimization of schema matching processes," in *Proc. 13th International Conference On Extending Database Technology (EDBT)*, 2010, pp. 453-464.
- [17] D. Lembo, M. Lenzerini, and R. Rosati, "Query rewriting for inconsistent DL-lite ontologies," in *Proc. the 5th Int Conf on Web Reasoning and Rule Systems*, 2011, pp.155-169.

- [18] G. Q. Li, H. Zou, and F. Yang, "Fuzzy ontology and fuzzy D-S evidence theory based context modeling and uncertainty reasoning," *Journal of Convergence Information Technology*, vol. 6, no. 12, pp. 185, Dec 2011.



Cao Kening was born in 1982. He is a PhD candidate of Hunan University, a student member of China Computer Federation(CCF). His current research interests include Internet of Things, Complex Event Processing. etc.



Li Renfa was born in 1957. He is a Professor and PhD supervisor of Hunan University, a senior member of CCF. His main research interests include CPS, embedded systems and wireless sensor networks.



Wang Fengjuan was born in 1987. She is pursuing her Master degree in Hunan University. Her current research interests include Internet of Things, Complex Event Processing. etc.