TELKOMNIKA, Vol. 11, No. 8, August 2013, pp. 4757~4765
e-ISSN: 2087-278X

A Conflict Context Reasoning Method based on Dempster-Shafer Theory in Ubiquitous Computing

Xinkai Yang
Computer Dept., College of Information Mechanical and Electronic Engineering, Shanghai Normal University Shanghai 200234, China
e-mail: xkyang@shnu.edu.cn

Abstract

In this paper, one conflict context reasoning method based on Dempster-Shafer theory is proposed. Firstly the context conflict problems are illustrated and partitioned based on theory of evidence. Then the context model combined with Dempster-Shafer theory is presented and applied to the reasoning method based on Dempster rule of combination. The effectiveness of this method is verified with a RFID application example.

Keywords*: context reasoning, Dempster-Shafer theory, ubiquitous computing*

1. Introduction

The vision of ubiquitous computing environments is characterized with a spectrum of computation and communication enabled sensing devices that can seamlessly augment human thoughts and activities [1]. The application running on these devices adjusts its behavior based on the environmental information and it is named as a context-aware application. The pieces of interesting environmental information are regarded as contexts. For example, context-aware applications adapt according to the location and time of use, the collection of nearby people and the accessible devices, as well as changes to those objects over time.

Context acquisition and context management are the most important requirements for context-aware systems. Usually context management system frameworks are divided in different conceptual layers. The whole tasks of context management can be assigned to the corresponding layers [2], including collecting raw sensor data from lower level, extracting and reasoning high level context information, aggregating and storing context information after eliminating wrong context, and then providing this information to the interested applications and users. In developing context-aware systems, the ability to model and consistently reason with high level contexts at the semantic level are the most important tasks. Some new semantic techniques with well-defined standards and ontology are required to deal with context at the semantic level [3]. Then it is able to handle the dynamics of environment intelligently in context-aware system.

However, during the execution of reasoning process, different conflicting situations can arise due to environmental noises. Contexts available to these applications may be abnormal or imprecise. This may result in context inconsistencies, which mean that contexts conflict with each other. These conflicting situations may set the application into a wrong state or affect the capability to adapt to the evolving situation. Some earlier works have been launched to handle these problems, but they are limited to specific context elements and specific scenarios. For example, reference [4] uses some simple strategies such as drop all, drop last, drop first to handle inconsistent context automatically. Sometimes users are involved to resolve context conflicts [5], or the conflict mediation is taken on the basis of some predefined static policies [6]. These strategies may distract user, or discard some important context objects because of the absence of semantic meanings.

In this paper, we try to provide one solution to deduce high level context from the low level conflict data based on Dempster-Shafer theory. In ubiquitous computing environment, the data is coming from multiple sources. These multiple sources provide different assessments for the same frame of discernment. Dempster-Shafer theory is based on the assumption that these

Received February 25, 2013; Revised May 24, 2013; Accepted June 3, 2013

sources are independent. And this assumption is compatible with the characteristics of ubiquitous computing.

The main contribution of the paper is twofold: (i) It analyzes the application feasibility of the theory of evidence in ubiquitous computing environment. In particular, it shows the use of Dempster-Shafer theory in context model to support reasoning process. (ii) It simplifies the calculation of distance function by using Hamming distance in the reasoning process. The rest of this paper is organized as follows: In section 2, the conflict context problem is illustrated with RFID examples and the feasibility of Dempster-Shafer theory is discussed. Dempster-Shafer theory and Dempster rule of combination are introduced in section 3. In section 4 the context model combined with Dempster-Shafer theory and the reasoning mechanism are presented. Experimental results are provided and analyzed in section 5. The conclusions and future work are discussed in section 6.

2. The Conflict Context Problem and Analysis

In this section, a warehouse management system is provided to illustrate the phenomena of context inconsistency in details. We argue that theory of evidence is one promising way to deal with the context inconsistency problem and context-aware applications can benefit from the detection and elimination of inconsistent context.

2.1. The Missed-Read and Cross-Read Problems

The RFID technology is widely applied in ubiquitous computing, such as warehouse and supply chain management, healthcare, etc. One of the major reasons is its ability to identify RFID tags physically covered by the other objects. The RFID technology comprises three components: antennae, readers, and tags. Through antenna communications, readers may track goods attached with tags.

Consider an RFID-enhanced warehouse management scenario in reference [11]. In a warehouse, a forklift is responsible for moving cases from a loading dock to a packaging site. To support automatic tracking of cases, the forklift is equipped with an RFID reader, and the cases are also labeled with RFID tags. Furthermore, there is another RFID reader installed at the packaging site to read tags of any cases that reach the packaging site. Suppose that each reading at a reader generates a context about the location of a tracked case. Ideally, the set of contexts generated at the packaging site should match those generated at the loading dock before the transportation.

However, the perceived read rate (i.e., the percentage of tags in a reader's vicinity that are actually reported) in real-life RFID deployments may fall below 70%. This means that at least 30% of all contexts may have been lost. If the warehouse management system cannot catch this anomaly, the anomaly may lead to an incorrect inventory ledger.

A typical solution to the missed read problem from industry is to power up the readers or increases the sensing coverage of the antennae. This alleviates the missed read problem by making the affected readers more sensitive to radio signals and thus reducing their read miss rates. On the other hand, an adjusted RFID reader may incidentally read the other tags. This is generally known as the cross read problem. Therefore, applications should decide a good tradeoff between alleviating missed reads and reducing cross reads. In practice, the problems of cross reads can be affected by a number of factors, such as tag positions, inter tag distances, packaging, speed of production lines, detuned frequencies, human bodies, ambient radio frequency noises, humidity, and so on [4]. Many of these factors do vary across time. Therefore, the RFID parameters for readers need to be tuned continually so that applications may maintain a good balance between missed reads and cross reads. It also demands an efficient approach to ensuring context consistency or spotting out inconsistent context so that the application may respond adaptively.

2.2. Type of Data Sources

The above example shows that RFID readers with different sensing coverage may incidentally read the other tags and thus produce conflict data. Suppose there are four RFID readers with varying degrees of resolution: Reader 1; Reader 2; Reader 3; Reader 4. When they detect one same tag at the same time, the reading results can be summarized into the following two types: consistent data source (Figure 1) and arbitrary data source (Figure 2).

Consistent data source means that there is at least one element that is common to all subsets. From our target identification, this could look like:

Reader 1 detects a tag in vicinity A.
Reader 2 detects two tags: one in vicinity A and one in vicinity B.
Reader 3 detects two tags: one in vicinity A, one in vicinity C.
Reader 4 detects three tags: one in vicinity A, one in vicinity B, one in vicinity D.

Arbitrary data source corresponds to the situation where there is no element common to all subsets, though some subsets may have elements in common. One possible configuration in our target identification example:

Reader 1 detects one tag in vicinity A.
Reader 2 detects two tags: one in vicinity A and one in vicinity B.
Reader 3 detects one tag in vicinity C.
Reader 4 detects two tags: one in vicinity B, one in vicinity D.

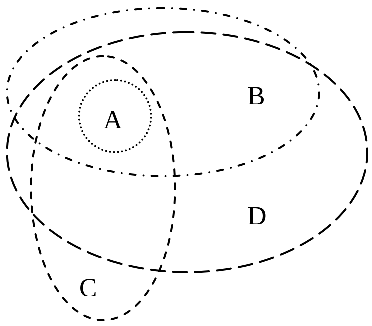

Figure 1. Consistent Data Source

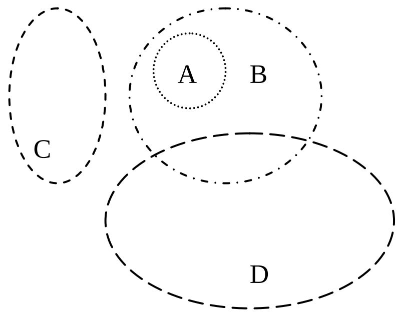

Figure 2. Arbitrary Data Source

From the view of theory of evidence, each of these two possible configurations of data from multiple sources has different implications on the level of conflict associated with the situation. In the case of arbitrary data source, there is some agreement between some sources but there is no consensus among sources on any one element. Consistent data source implies an agreement on at least one evidential set or element. Traditional probability theory cannot handle these types of evidence without resorting to further assumptions of the probability distributions within a set, nor can probability theory express the level of conflict between these evidential sets. Dempster-Shafer theory is a framework that can handle these various evidentiary types by combining a notion of probability with the traditional conception of sets. In addition, in Dempster Shafer theory, there are many ways which conflict can be incorporated when combining multiple sources of information.

The Dempster-Shafer theory, also known as the theory of belief functions, allows us to base degrees of belief for one question on probabilities for a related question [15]. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ from probabilities will depend on how closely the two questions are related.

3. Dempster-Shafer Theory

3.1. Three Functions

Dempster-Shafer theory is a mathematical theory of evidence, which was originally proposed by Dempster in 1967 [15] and was expanded by his student Shafer in 1976 [16]. In a finite discrete space, Dempster-Shafer theory can be interpreted as a generalization of probability theory where probabilities are assigned to sets as opposed to mutually exclusive singletons [17]. This theory came to the attention of AI researchers in the early 1980s, when they were trying to adapt probability theory to expert systems. One of the most important features of Dempster-Shafer theory is that the model is designed to cope with varying levels of precision regarding the information and no further assumptions are needed to represent the information. It also allows for the direct representation of uncertainty of system responses where an imprecise input can be characterized by a set or an interval and the resulting output is a set or an interval.

In the theory of evidence, the sample space is also known as frame of discernment which will be represented with Θ. Θ contains all the objects that are mutually exclusive, i.e. $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$. These objects represent the evidence from different sources and can be used to get a degree of belief (represented by a belief function) in the reasoning system. There are there important functions in Dempster-Shafer theory, the basic probability mass, the belief function, and the plausibility function.

Definition 1. Basic probability mass (bpm)

The universal set Θ represents all possible states of a system under consideration. Then the power set $P(\Theta)$ is the set of all subsets of Θ, including the empty set Φ. The elements of the power set can be taken to represent propositions concerning the actual state of the system by containing the states in which the proposition is true. In the theory of evidence, each element of the power set will be assigned a basic probability mass.

$$m : P(\Theta) \to [0,1] \tag{1}$$

The value of the bpm for a given set A (represented as m (A)) expresses the proportion of all relevant and available evidence that supports the claim that a particular element of Θ belongs to the set A but to no particular subset of A [19].

The definition of the basic probability mass satisfies:

$$m(\phi) = 0\,, \tag{2}$$

$$\sum_{A \in P(\Theta)} m(A) = 1\,, \tag{3}$$

Equation (2) indicates that the basic probability mass of the empty set is zero. Equation (3) indicates that the bpm of all the members of power set add up to a total of 1.

From the definition of basic probability mass, we can get the following two functions. There are:

Definition 2. Belief function

The belief function for a set A is defined as the sum of all the basic probability mass of the proper subsets (B) of the set of interest (A) ($B \subseteq A$). The value of belief function represents the total belief assigned to the set A.

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{4}$$

And,

$$Bel(\Phi) = 0 \tag{5}$$

$$Bel(P(\Theta)) = 1 \tag{6}$$

Equation (5) means that the total belief of the empty set is zero. Equation (6) means that the total belief of the power set is equal to 1.

Definition 3. Plausibility function

The plausibility function is the sum of all the basic probability mass of the set B that intersects the set of interest A ($B \cap A \neq \Phi$).

$$Pl(A) = \sum_{B \cap A \neq \Phi} m(B) \tag{7}$$

Obviously, $Bel(A) \le Pl(A)$. The interval $[Bel(A), Pl(A)]$ represents the uncertain interval for the proposition (set) A. The interval $[0, Bel(A)]$ represents the completely trustful interval. The interval $[0, Pl(A)]$ represents the undoubted interval as the proposition A is true.

3.2. The Dempster Rule of Combination

Often used as a method of sensor fusion, Dempster–Shafer theory is based on two ideas: obtaining degrees of belief for a related question, and combining such degrees of belief using Dempster's rule or other augmented combination rules. Different rules of combination have different assumptions about the data. Dempster's rule is based on the assumption that all the data sources are independent.

Dempster’s rule combines multiple belief functions through their basic probability masses. These belief functions are defined on the same frame of discernment, but are based on independent arguments. Specifically, the combination (called the joint mass *m_{12}*) is calculated from the two sets of masses m_1 and m_2 in the following manner:

$$m_{12}(A) = {}^{1}\!/_{1-K} \sum_{B \cap C = A \neq \Phi} m_1(B) m_2(C) \tag{8}$$

$$m_{12}(\Phi) = 0 \tag{9}$$

$$K = \sum_{B \cap C = \Phi} m_1(B) m_2(C) \tag{10}$$

In Equation (8), K represents basic probability mass associated with conflict. The number K is determined in Equation (10) by summing the products of the basic probability masses of all sets where the intersection is null. The normalization factor (1-K) in Equation (8) has the effect of completely ignoring conflict and attributing any probability mass associated with conflict to the null set [20].

4. Conflict Context Reasoning Based on Dempster–Shafer Theory

Context from RFID readers, sensors, as well as other sensing devices is considered as evidence in the domain of theory of evidence. This evidence can be synthesized to get the belief of different propositions with Dempster combination rules. The high level context comes from the properly reasoning result of these propositions. But the context structure and their relationship cannot be fully expressed with the theory of evidence. One ontology-based context model is provided here to fully exploit the advantage of the theory of evidence (Figure 3).

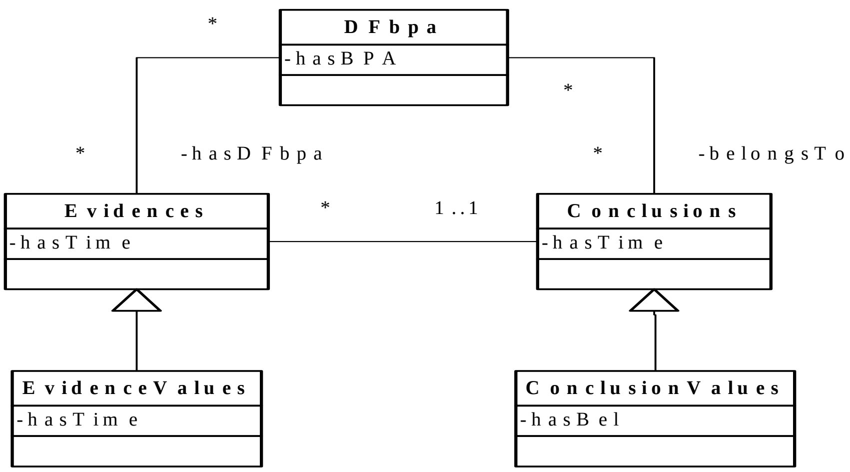

Figure 3. Context Model base on DS Theory

In this model, the class *Evidences* represents the evidence, the class *Conclusions* represents the conclusions. These two classes correspond to the sample space and the power set in the theory of evidence respectively. On the other hand, these two classes also correspond to the low-level context and high level context in the reasoning process. They all have one data attribute - *hasTime* to record the context acquisition time, which can be used to represent the freshness of the context. The subclass *EvidenceValues* of class *Evidences* represents the values of evidence. The subclass *ConclusionValues* of class *Conclusions* represents the values of conclusions. The data attribute *hasBel* of subclass *ConclusionValues* represents the value of belief function and reflects our confidence in this conclusion.

From the definition of basic probability mass (*bpm*) in Dempster-Shafer theory we know that it expresses the proportion of an evidence to support its corresponding conclusion (proposition). One bpm exists between the evidence and its corresponding conclusion. Therefore, one new class DFbpa is introduced in the above conceptual model, which has the data attribute-*hasBPA* to represent the basic probability mass. Two classes-*Evidences* and *Conclusions* are linked together with the class DFbpa.

Based on the above context model, we can begin to reason high level context from sensor data using Dempster rule of combination defined in section 3.2. However, as we have pointed out in the former section, some conflicts may arise among the evidence provided by different sensors. These conflicts will lead to the information fusion results contrary to the real fact. In Equation (10), it means all the evidence conflict completely when k=1. Dempster rule of combination cannot be applied in this conflict situation. So, a number of methods and combination operations have been developed to address this problem posed by strongly conflicting evidence. In this paper the concept of distance function is introduced, which is firstly proposed in reference [21]. The distance function is used to measure the degree of evidence's similarity. Furtherly the degree of one single evidence supported by other evidence is obtained. The degree of support is used as the weight of evidence in Dempster rule of combination. In ref. [21] the basic probability mass is represented as a coordinate vector in the coordinates of power set, and then the distance function is calculated as:

$$d(\overline{m_1}, \overline{m_2}) = \sqrt{(\| \overline{m_1} \|^2 + \| \overline{m_2} \|^2 - 2 < \overline{m_1}, \overline{m_2} >)/2} \qquad (11)$$

where $\| \overline{m} \|^2 = < \overline{m}, \overline{m} >$, $< \overline{m_1}, \overline{m_2} >$ is the inner product of two vectors and is calculated as,

$$< \overline{m_1}, \overline{m_2} > = \sum_i \sum_j m_1(A_i) m_2(B_j) \frac{|A_i \cap B_j|}{|A_i \cup B_j|} \qquad (12)$$

So, the procedure of context reasoning algorithm based on theory of evidence includes the following steps.

1) To determine the evidence space and conclusion framework according to the established domain ontology. That is to create the *Evidences* class and the *Conclusions* class. Then their subclasses *EvidenceValues* and *ConclusionValues* are created. The relevant classes are instantiated depending on the conceptual model.

2) To assign the basic probability mass of evidence for the corresponding proposition and evaluate the attribute *hasBPA* of class *DFbpa*. Then the conflict factor k is calculated.

3) Set a thresholdε, if k is less than ε, the Dempster rule of combination rule can be applied directly for evidence fusion; otherwise, if k is not less than ε, the distance function is calculated to act as the weight of evidence, then Dempster rule of combination is used to get the final belief value of evidence.

4) To calculate the attribute *hasBel* value of *ConclusionValues*, the maximum one is the reasoning result.

5. Experiment Results and Discussion

We use one application scenario of conveyor belt described in reference [22] as the motivating example to verify the effectiveness of our provided method. In this example the sensors' signal strength is regarded as low level context and package's position is deduced as high level context. The details of this example are as follows.

Along with a conveyor belt carrying packages (each of them is linked with one RFID tag) to pass through an inspection zone at an airport, four RFID readers are deployed to partition the inspection zone into a series of segments, and they are labeled as reader0, reader1, reader2 and reader3, and their positions are denoted as 0, 1, 2, and 3, respectively. When a package passes through the inspection zone of each RFID reader, its position is sensed and calibrated according to the position of the reader that receives the strongest signal strength(We name this kind of result as Max signal position, see Table 1). Due to the missed reads and cross reads problems described in section 2.1, this method cannot always produce the ideal results. Table 1 shows the position of the package deduced from different methods.

Table 1. Package's Position Deduced from Different Methods

Sample Fragment	*Frame of Sample*			*Dynamic position*	*Max signal result*	*Revised result*	*Distance function*
	frame 1	frame 2	frame 3				
Case 1	(30, 20, 5, 10)	(30, 40, 5, 10)	(30, 40, 5, 40)	0→1→3	3	3	2
Case 2	(20, 40, 20, 5)	(10, 25, 30, 10)	(20, 30, 20, 15)	1→2→1	1	2	8
Case 3	(30, 20, 10, 20)	(30, 20, 15, 30)	(40, 25, 40, 35)	0→0→2	2	2	6
Case 4	(20, 30, 20, 10)	(10, 15, 30, 20)	(8, 35, 30, 22)	1→2→1	1	2	7

In a real environment, multiple context stream fragments may be available. We randomly pick one of such sample stream fragments to get one sample case. We denote four readers' signal strength as the tuple (r0, r1, r2, r3) and name it as a sample frame. In order to determine a package's position, we consecutively sampling at a fixed time interval and form a sample stream produced by the four sensors. To simplify our discussion, each sample stream fragment consists of three sample frames.

Because the signal strength from different sources may conflict each other, Dempster rule of combination cannot be applied directly in this situation. The revised rule based on Equation (11) is used to deduce the package's position. To simplify the calculation, the distance function is replaced with the total sum of the Hamming distance of a pair of consecutive sample frame in one sample stream fragment.

Take case 4 for example: The Hamming distance between the first pair of two sample frame is 4 because the value of r0 should be edited from 20 to 10 (i.e., one change), and the values for r1, r2, and r3 alike. In the same manner, the Hamming distance of the next pair of sample is only 3 because the value for r2 does not change. Summing up these two values gives a value of 7, which is shown in the "Distance function" column in Table 1.

The effectiveness of this method can be exhibited in this example. During its execution, the application accepts readings from location RFID readers as parametric inputs. When conflict context is detected, the main component uses a context dropping strategy to handle (i.e., drop) inconsistent package position values. The removal of value is substituted with the former position value. For instance, from Table 1, we observe that execution of the revised result of case 4 produces 2 as the position result. This is because the application has detected the abnormal value of its distance function and removed the last position value (which is 1) produced by the maximum signal strength method.

The above results achieved from the position detection experiment carried out in the conveyor belt scenario. Furthermore we compare the performance of our approach to other similar method, e.g. reference [23], which presents a framework for realizing dynamic context consistency management. One inconsistency detection method based on a semantic matching and inconsistency triggering model is provided. Comparing our approach to this framework can not be done directly because the construction of the training context instances varies in terms of the granularity. We have implemented one similar test environment, in which a context source thread sent 2000 context instances to the framework at different rate (instances per second). The experiment results show that our approach outperforms several features used in other similar approaches. The accuracy is increased by 2.82%, the micro precision by 0.02 in comparison to the best values achieved by other approaches.

6. Conclusion

Due to sensors' error or environmental noises, different conflicting situations can arise in ubiquitous computing environment. To reason high level context from low level sensing devices' data for their effective and meaningful exploitation is increasingly becoming a challenging issue. It is necessary to develop some proper solutions to represent, pruning conflict data in ubiquitous computing environment. In this paper, the missed read and cross read problems are discussed thoroughly, the data sources are partitioned based on the theory of evidence. Then one method based on Dempster rule of combination is introduced to reason high level context. A context model combined with Dempster-Shafer theory is presented in this paper, which works as an effective model to support high level context reasoning process. The effectiveness of the revised method is verified by some experiments and comparison with other conventional approaches also shows its advantage.

There are several directions to improve in our work: 1) With respect to the context model combined with Dempster-Shafer theory, it is better to enrich and refine the model itself. 2) With respect to the reasoning method base on Dempster rule of combination, it is better to make it more flexible and suitable for computing environment. This could be very useful especially in the case of huge amount of data and highly dynamics in the sensor level. It is also important to develop more applications to test the effectiveness of this method in a variety of application scenarios. Other kind of sensing devices and algorithms could be employed to extract more information and improve the performance of our implementation.

Acknowledgments

We would like to thank the helpful suggestions from my colleagues in Shanghai Normal University. We would also like to thank the anonymous reviewers for their invaluable comments. This work is supported in part by a grant of Shanghai Normal University.

References

[1] Roman M, Hess C, Cerqueira R, Ranganathan A, Campbell H, Nahrstedt K. A Middleware Infrastructure for Active Spaces. *IEEE Pervasive Computing*. 2002; 1(4): 74-83.
[2] Matthias B, Schahram D, Florian R. A Survey on Context-aware Systems. *Int. J. Ad Hoc Ubiquitous Computing.* 2007; 2(4): 263-277.
[3] Aroyo M, Dimitrova V, Kay J. *Personalization on the Semantic Web: Editorial*. Proceedings of Workshop on Personalization on the Semantic Web. UK. 2005: 25-26.
[4] Chang Xu, Cheung SC, Chan K, Chunyang Y. *On Impact-oriented Automatic Resolution of Pervasive Context Iinconsistency*. Proceedings of the 6th Joint Meeting of European Software Engineering Conference and ACM SIGSOFT Symposium on The Foundations of Software Engineering. ACM. 2007: 569–572.
[5] Anind D, Jennifer M. Designing Mediation for Context-aware Applications. *ACM Trans. Comput.-Hum. Interact.* 2005; 12(1): 53–80.
[6] Insuk Park, Dongman Lee, Soon J Hyun. *A Dynamic Context-conflict Management Scheme for Group-aware Ubiquitous Computing Environments*. Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05). IEEE Computer Society. 2005; 1: 359–364.
[7] Licia Capra, Wolfgang Emmerich, Cecilia Mascolo. Carisma: Context-aware Reflective Middleware System for Mobile Applications. *IEEE Transactions on Software Engineering*. 2003; 29: 929–945.
[8] Ranganathan A, Al-muhtadi J, and Cambell H. Reasoning about Uncertain Contexts in Pervasive Computing Environments. *IEEE Pervas. Comput.* 2004; 3(2): 62–70.
[9] Ranganathan A, Cambell H. An Infrastructure for Context-awareness Based on First Order Logic. *Person. Ubiq. Comput*. 2003; 7: 353-364.
[10] Rauschmayer A, Knapp A, Wirsing M. *Consistency Checking in An Infrastructure for Large-Scale Generative Programming*. In Proceedings of the 19th International Conference on Automated Software Engineering. 2004: 238-247.
[11] Rao J, Doraiswamy S, Thakkar H, Colby S. A Deferred Cleansing Method for RFID Data Analytics. Proceedings of the 32nd International Conference on Very Large Data Bases. 2006: 175-186.
[12] Jeffery R, Garofalakis M, Frankin J. *Adaptive Cleaning for RFID Data Streams*. Proceedings of the 32nd International Conference on Very Large Data Bases. 2006: 163-174.
[13] Yau S, Karim F. An Adaptive Middleware for Context-sensitive Communications for Real-time Applications in Ubiquitous Computing Environments. *Real-Time Syst.* 2004; 26(1): 29-61.

[14] Bryan Scotney, Sally McClean. Database Aggregation of Imprecise and Uncertain Evidence. *Information Sciences-Informatics and Computer Science*. 2003; 155(3): 245-263.
[15] Shafer Glenn. Perspectives on the Theory and Practice of Belief Functions. *International Journal of Approximate Reasoning.* 1990; 3: 1-40.
[16] Shafer Glenn. A Mathematical Theory of Evidence. Princeton University Press. 1976.
[17] Binaghi L, et al. Slope Instability Zonation: A Comparison Between Certainty Factor and Fuzzy Dempster-Shafer Approaches. *Natural Hazards*. 1998; 17(1): 77-97.
[18] Yager R. Arithmetic and Other Operations on Dempster Shafer Structures. *International Journal of Man-Machine Studies*. 1986; 25: 357-366.
[19] Klir J, Smith M. On Measuring Uncertainty and Uncertainty-based Information: Recent Developments. *Annals of Mathematics and Artificial Intelligence*. 2001; 32: 5-33.
[20] Yager R. On the Dempster-Shafer Framework and New Combination Rules. *Information Sciences*. 1987; 41: 93-137.
[21] Jousselme A, Grenier D, Bosse E. A New Distance Between Two Bodies of Evidence. *Information Fusion*. 2001; 2(2): 91-101.
[22] Lu H, Chan K, Tse H. *Testing Pervasive Software in the Presence of Context Inconsistency Resolution Services*. in Proceedings of 30th International Conference on Software Engineering (ICSE 2008). 2008: 100-110.
[23] Chang Xu, Cheung C, Chan K. *Inconsistency Detection and Resolution for Context-aware Middleware Support*. Proceedings of the 10th European Software Engineering Conference. ACM. 2005: 336-345.
[24] Hong Sun, Shi-ping Chen, Li-ping Xu, Ying-ying Chen. Base-on Cloud Computing A new type of distributed application server system design. *Indonesian Journal of Electrical Engineering*. 2012; 10(7): 1800-1807.
[25] Hermawan Hermawan, Riyanarto Sarno. Developing Distributed System With Service Resource Oriented Architecture. *Indonesian Journal of Electrical Engineering*. 2012; 10(2): 389-399.