Advances in Computer Science Research, volume 88
International Conference on Computer, Network, Communication and Information Systems (CNCI 2019)

Ontology Based Reasoning Rules for Target Tracking in WSNs

Jing Zhang[1, a], Qingxu Xiong[1, b]
[1]Beihang University, Beijing, China
[a]jingzhang1993@163.com, [b]qxxiong@buaa.edu.cn

Abstract. Wireless sensor networks (WSNs) are widely used for target tracking. For different tracking tasks, appropriate tracking methods should be deployed properly with WSNs according to various task environment and requirements. In this work, ontology based reasoning rules for target tracking are proposed. Unlike the traditional logical operator based reasoning rules, rules with the aid of ontology and behavior tree are constructed, which have formulated the complex reasoning of the target tracking. Through the example of application of WSNs on high-mobility air target tracking, feasibility of the proposed method has been proved.

Keywords: WSNs, target tracking, ontology, reasoning rules.

1. Introduction

Along with the development of semantic network and artificial intelligence, WSNs based target tracking technique has been drawing more and more attention [1]-[3]. However, since the data process of each node in WSNs needs to be highly automatic and intelligent, the single type of target tracking algorithm for WSNs may not be able to meet the various task requirements. For example, the traditional Kalman filter method could accurately track target that in the uniform rectilinear motion [4]. However, it suffers poor performance when the target had large maneuver. Moreover, the particle filter algorithm has high accuracy in tracking target with large maneuver [5]. However, the algorithm has high computation complexity and high energy consumption, so that it may not be suitable for sensor nodes [6]. Therefore, it is necessary for WSNs to be able to process the data intelligently according to the task situation. Reasoning ruling is one of possible approach to the problem. However, traditional reasoning rules are usually on the basic of logical operation, which may not be sufficient for a node to deal with the complex situation of task [7]-[9].

To address these issues, a set of reasoning rules based on ontology and behavior tree is proposed, to make the node choose the appropriate algorithm intelligently according to different task situations. In this method, the ontology of task is constructed, node and target tracking method, which enabled each node in WSNs gaining the knowledge of task situation, task environment, target motion characteristics and the other factors that have influence on the target tracking algorithm. And based on the built ontology, the behavior tree is utilized to establish the reasoning rules. According to these rules, the node in WSNs can automatically select suitable tracking algorithms to achieve the accuracy of target tracking

In the following sections, the ontology constructed will be firstly demonstrated to show the task, target, node and related algorithm. Then the mechanism of the reasoning rules of target tracking will be explained. In the last section, an example of reasoning of air target tracking is given to show the feasibility of proposed method.

2. Construction of Reasoning Rules of Target Tracking in WSNs

2.1 Construction of Ontology for Reasoning

To formulate the reasoning of target tracking algorithm, we construct the condition ontology and result ontology, which respectively represent the known information and results of the reasoning. A modified Methontology method is employed to construct the above mentioned ontologies. And the work flow of our method is shown in Fig.1. It is can be seen that our method consists of two stages,

Copyright © 2019, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

the result ontology construction stage and the condition ontology construction stage. In the first stage, the result ontology, namely the ontology of target tracking algorithm, is constructed by use of a simplified Methontology framework [4], where the construction of axiom, constants and instance are omitted. Then, in the second stage, all the concepts and elements that influence the performance of the target tracking algorithm are listed. Based on this list, the condition ontology is established by the same approach as the result ontology. The work flow of the method is shown in Fig. 1.

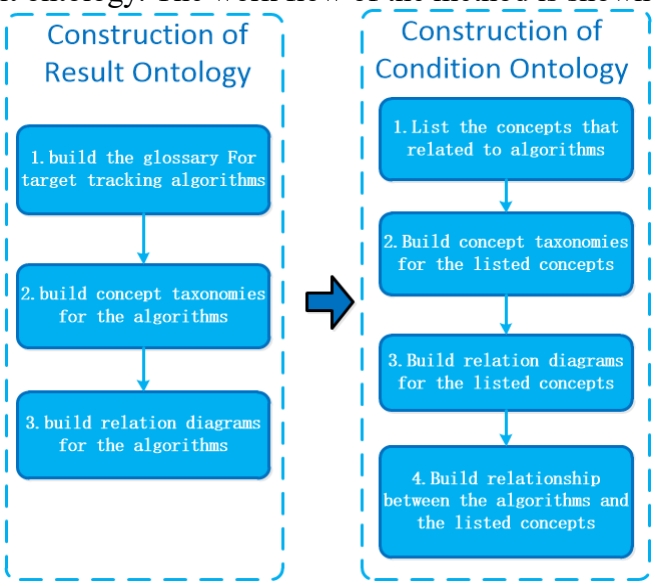

Fig. 1 Work flow of the construction of ontology

The structure of the ontology of target tracking algorithm is demonstrated in Fig. 2. As we can see, it consists of two parts: the data association process and the estimation of target status, respectively, which are two major steps of the target tracking algorithm. The algorithm of data association process, which filters out measurement from the target, contains two key issues that are the range gate and the association matrix calculation. According to the characteristics of different methods, the taxonomies of the range gate and association matrix calculation method are made to formulate the data association algorithm reasoning. As for the target status estimation, the ontology of the target motion model and the status estimation method are constructed, respectively. The motion models are listed as the sub-ontology of the target motion model ontology. Moreover, similar with the association matrix calculation method, the method to estimate the taxonomies of the status are established based on the characteristics of the algorithms.

The condition ontology is made up of three parts: that is, task contents, node ability and target property. There are two types of concepts of the task. One of them is the factors that have requirements for performance, including the required tracking accuracy and the number of potential targets. The other one is the elements that may have impact on the performance of algorithm, such as task environment and task time. As for the node ability, it has three concepts. The sensor type specifies the characteristics of target measurement of the node, and the data processing ability and the energy-level indicating resources which the node could provide to the algorithm. The target property provides the real-time information of the target, so that the node could select the most suitable algorithm accordingly. As we can see, it consists of location, velocity, maneuver type and current tracking error. The location represents the current estimated target location in space. The velocity has three attributes, which are the maximum velocity, the minimum velocity and the current velocity. As for the maneuver type, in the work, the targets whose maximum acceleration is larger than the gravitational acceleration are classified as high mobility target, such as battle plane and vehicle. And the targets whose maximum acceleration is less than the gravitational acceleration are classified as low mobility target, including ships and airliner. The tracking error represents the current error of target status estimation.

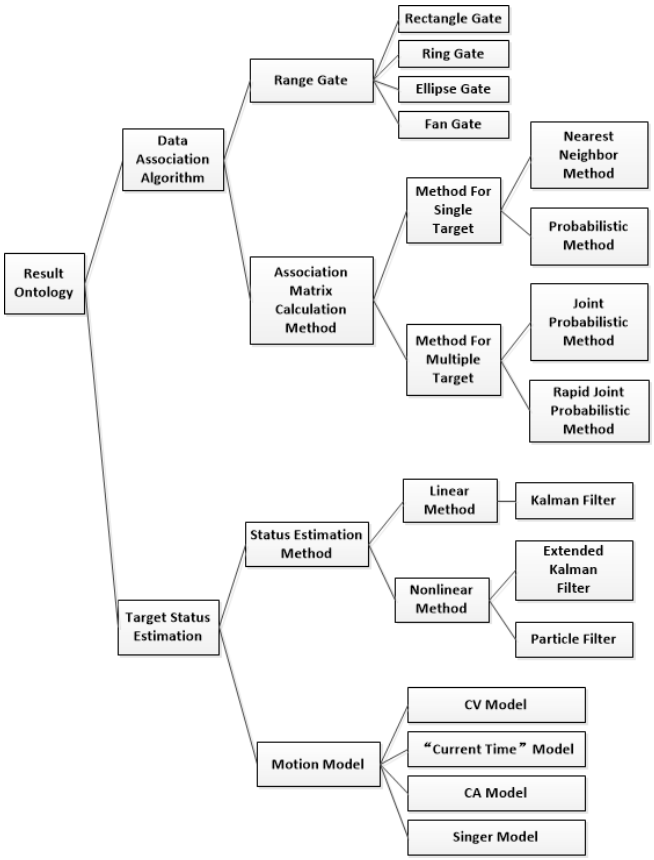

Fig. 2 The Structure of result ontology

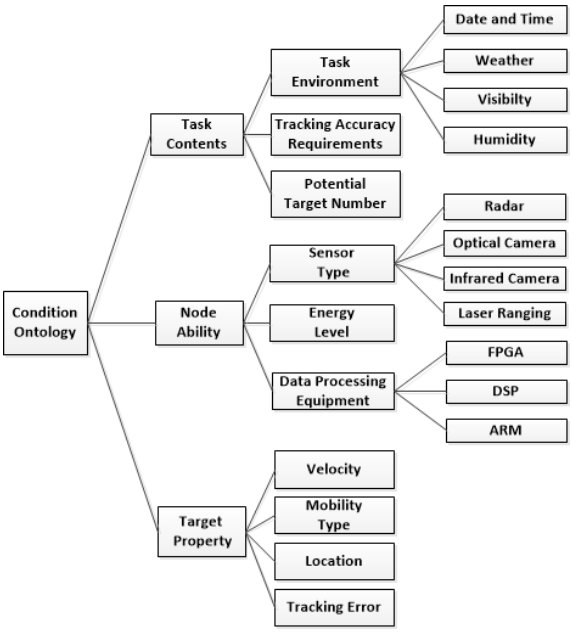

Fig. 3 The Structure of condition ontology

It is worth mentioning that the relationship between the condition ontology and result ontology with the object property is built, which is the key to establish the reasoning rules. Therefore, the relation between the condition ontology and result ontology will be explained in detail when the illustration of establishment of the reasoning rules is given.

2.2 The Establishment of Reasoning Rules

According to the ontology mentioned above, we can see that there are different kinds of algorithms of the target tracking technique. Actually, these algorithms have different performance under different task situation. In order to make the best use of these algorithms, a set of reasoning rules is built to

make the node select the most appropriate target tracking strategy according to the current task situation. Since the target tracking technique contains two major sub-processes, the data association and target status estimation, reasoning rules for these two sub-processes are established respectively.

In data association process, the accuracy mainly relies on the selection of range gate and the performance of association matrix calculation method. To start with, the reasoning rules are built for the selection of range gate. The rules are built as a function as follows:

$$RangeGate = \text{hasRangeGate}(S_{target}, A_{Node}, T_{TargetMotion}). \tag{1}$$

Here, S_{target} is the status of target. A_{Node} and $T_{TargetMotion}$ represent the node ability and target motion, respectively. HasRangeGate () stands for the reasoning function, in which the reasoning is performed according to the behavior tree built on the basis of the relationship between the range gate ontology and task situation ontology. The structure of behavior tree is shown in Fig. 4. As we can see, when the target is not steadily tracked, the ring gate will be selected as the most optimized choice. As soon as the target is tracked, for the nodes that have radar, the fan gate is much more appropriate because the radar has low accuracy in tangential direction. As for the other nodes, they usually utilize the rectangle gate or ellipse gate depending on the velocity of the target.

For the association matrix calculation method, a reasoning function is also constructed, as shown in Fig. 5. According to the built behavior tree, the reasoning function of association matrix calculation method is designed as below:

$$Method_{Association} = \text{hasAssociationMethod}(N_{Target}, E_{Task}). \tag{2}$$

Here, N_{target} refers the number of the potential targets. E_{Task} denotes the environment of task. HasAssociationMethod() stands for the reasoning function. And the reasoning process in this function is given by a behavior tree which is demonstrated in Fig. 5. As it shows, if it indicates that there may be more than one target in the task area, then the joint probabilistic association method will be applied. And if the task is for single target tracking, then the node needs to choose method according to the task environment [8]. To be specific, for the environment where weather is sunny and visibility is good, the nearest neighbor method will be the most appropriate, since this method is very simple and could save a lot of energy for the node. And when the weather is foggy or rainy, then the probabilistic method will be applied, because this method has strong robustness to the noise. It is worth mentioning that, for the multiple-target task, the joint probabilistic method has good performance, but this method has high computational complexity as well, which is too energy-consuming for the node in WSNs [9]. The above two reasoning rules are built for the data association process. According to these two rules, the node can automatically select the appropriate data association method, which provides the data foundation for accurate target tracking.

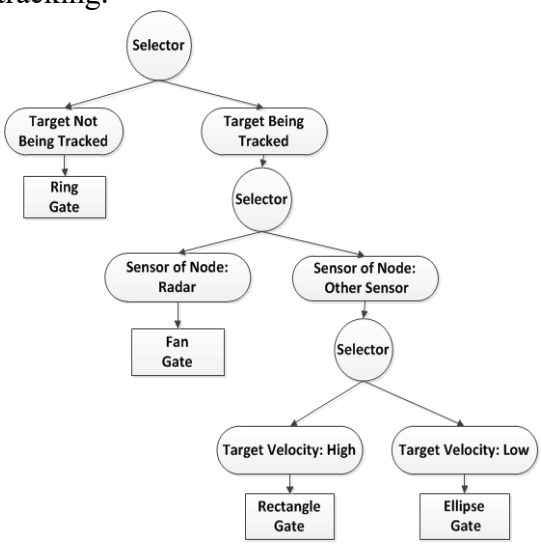

Fig. 4 The structure of behavior tree for range gate reasoning

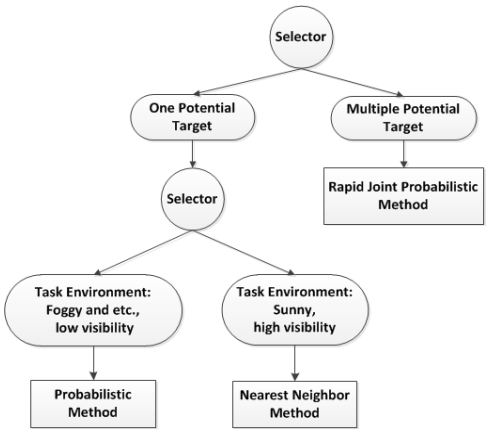

Fig. 5 The structure of behavior tree for the association matrix calculation method reasoning

The estimation of target status is the most critical process in target tracking. And the target motion model and the target estimation methods are two major factors that affect the accuracy of the estimated target status. Therefore two reasoning rules are designed for the selection of motion model and status estimation method [10]-[12]. To find the most appropriate target motion model, a function is built, which outputs the selected target motion model according to the target status, as follows:

$$Model = \text{hasMotionModel}(A_{Target}, S_{TargetMobility}). \tag{3}$$

Here, A_{Target} stands for the acceleration of target, which is a data attribute of target acceleration ontology. $S_{TargetMobility}$ denotes the status of target mobility, namely maneuvering or in steady movement. Has MotionModel() represents the reasoning function. This function is formulated based on the reasoning function. The motion model is selected according to a behavior tree designed based on the relationship between the motion model and input ontology, which is shown in Fig.6. As we can see, the selection of target motion model is based on the acceleration and the mobility status of the target. For the steadily moving target, the CV motion model is applied. And when the target has maneuver and the acceleration of the target is larger than the gravity acceleration which equals to 9.8m/s2, then the "Current Time" motion model will be used. And for the maneuvering target whose acceleration is less than 9.8m/s2, the CA motion model will be applied [10].

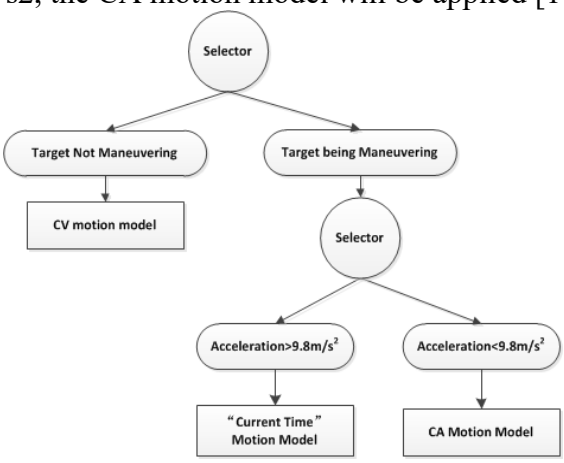

Fig. 6 The structure of behavior tree for motion model reasoning

The method of target status estimation outputs the estimated target current status and the predicted target future status, which has direct impact on the final target tracking. Similar to previous steps, a function of the reasoning is built as below:

$$Model = \text{hasMotionModel}(A_{Target}, S_{TargetMobility}). \tag{3}$$

Here, Node refers to the ontology of node ability. $S_{TargetMobility}$ denotes the status of target mobility. Has TrackingMethod() denotes the reasoning function. In this function, the reasoning process is performed on the basis of a behavior tree, which is shown in Fig. 7. According to this behavior tree, node would select kalman filter for the target whose acceleration is less than the gravity acceleration [13]-[15]. And when the acceleration of the target is larger than the gravity acceleration, selection of

the target status estimation method depends on the computation ability of node. For the node that has high-performance computation device, such as FPGA or DSP, the particle filter will be applied. If the node doesn't have enough computation power, then the extended kalman filter is going to be utilized.

3. Example of Target Tracking Algorithms Reasoning on WSN

To illustrate the feasibility of the method, an example of high-mobility air target tracking is introduced. There are two types of nodes deployed in the network. The first type is airborne node that is equipped with radar, FPGA, and fuel tank. The second type is unmanned ground vehicle (UGV) based nodes that have CCD camera, laser ranging devices, and ARM. Before the reasoning of the tracking algorithms, the instance of the condition ontology was built for this task, which is shown in Fig.8. As we can see, the definition of the attributives of instance is given by linking them with the dashed line. With the aid of these attributives, detailed information of the task situation can be collected. In the instance given, some of the information was defined before the task started, including task contents, ability of the airborne node and ground node. In this task, the weather condition is sunny to rainy, which leads to the visibility ranging from high to low. As for other information of the instance can only be given when the target is detected, such as the velocity of the target, the location of target and the tracking error of target.

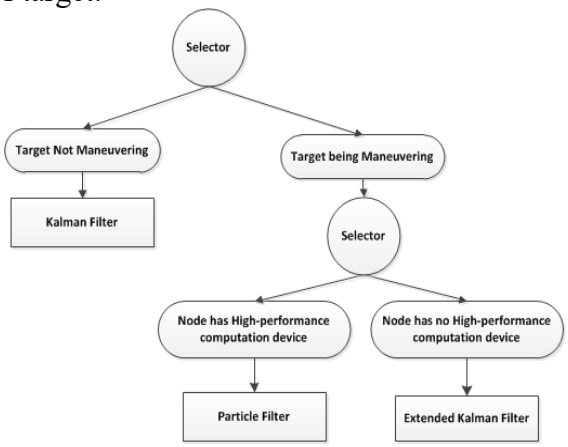

Fig. 7 The structure of behavior tree for target status estimation method reasoning

For the high-mobility air target tracking, tracking algorithms should vary according to different situations. Therefore, the reasoning result of two type's nodes will be demonstrated under three typical situations: that is, target moves steadily and is not tracked under good weather condition, target moves steadily and is being tracked under bad weather condition, and target maneuvers fiercely and is being tracked under bad weather condition.

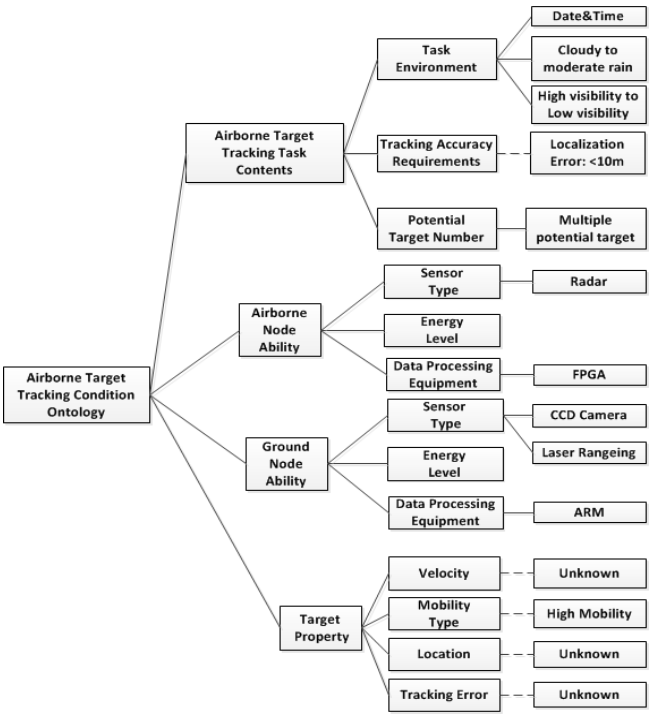

Fig. 8 The instance of condition ontology for high-mobility air target tracking

1) Situation: S_{targe}= Not Tracked, $T_{TargetMotion}$ = high mobility, E_{Task}= Sunny, A_{Target}=0. The reasoning result for airborne node is as follows:

1. Ring Gate = hasRangeGate(S_{target}, $A_{AirborneNode}$, $T_{TargetMotion}$)
2. NNM = hasAssociationMethod(Single, Sunny)
3. CV Motion Model = hasMotionModel(0,high-mobility)
4. Kalman Filter = hasTrackingMethod($A_{AirborneNode}$, No)

And the result of the reasoning for the ground node is shown below:

1. Ring-Gate = hasRangeGate(S_{target}, $A_{GroundNode}$, $T_{TargetMotion}$)
2. NNM = hasAssociationMethod(Single, Sunny)
3. CV Motion Model = hasMotionModel(0,high-mobility)
4. Kalman Filter = hasTrackingMethod($A_{AirborneNode}$, No)

2) Situation: S_{target}= Being Tracked, $T_{TargetMotion}$ = high mobility, E_{Task}= Rainy, A_{Target}=0. The reasoning result for airborne node is as follows:

1. Fan-Gate = hasRangeGate(S_{target}, $A_{AirborneNode}$, $T_{TargetMotion}$)
2. JPM = hasAssociationMethod(Single, Rainy)
3. CV Motion Model = hasMotionModel(0,high-mobility)
4. Kalman Filter = hasTrackingMethod($A_{AirborneNode}$, No)

And the result of reasoning for the ground node is shown below:

1. Rect-Gate = hasRangeGate(S_{target}, $A_{AirborneNode}$, $T_{TargetMotion}$)
2. JPM = hasAssociationMethod(Single, Rainy)
3. “CurrentTime”Time = hasMotionModel(A_{Target}, high-mobility)
4. Extended Kalman Filter = hasTrackingMethod($A_{AirborneNode}$, Yes)

3) Situation: S_{target}= Being Tracked, $T_{TargetMotion}$ = high mobility, E_{Task}= Rainy, A_{Target}>9.8m/s2. The reasoning result for airborne node is as follows:

1. Ellipse-Gate = hasRangeGate(S_{target}, $A_{GroundNode}$, $T_{TargetMotion}$)
2. JPM = hasAssociationMethod(Single, Rainy)
3. CV Motion Model = hasMotionModel(0,high-mobility)
4. Kalman Filter = hasTrackingMethod($A_{AirborneNode}$, No)

And the result for the ground node is shown below:

1. Fan Gate = hasRangeGate(S_{target}, $A_{AirborneNode}$, $T_{TargetMotion}$)
2. JPM = hasAssociationMethod(Single, Rainy)
3. "CurrentTime"Time = hasMotionModel(A_{Target}, high-mobility)
4. Particle Filter = hasTrackingMethod($A_{AirborneNode}$, Yes)

According to the above result, it reveals that for the first situation, which is typical in the beginning of the task tracking, the airborne node and the ground node applied the same target tracking algorithms. And when the weather condition turns bad, both the airborne nodes changed their tracking strategy to alleviate the negative effect brought by the weather on the target tracking. And in the last situation, the target tried to get rid of the tracking by fierce maneuvering. Then the airborne nodes and ground nodes automatically utilize different tracking algorithms according to the node ability to make sure the accurate target tracking can be achieved.

4. Discussion and Conclusion

In this paper, a novel method for target tracking algorithms reasoning for WSNs is proposed. This method could make the nodes in WSNs automatically select the most appropriate algorithms according to task situation, the ability of the node and target status, which will benefit the final accuracy of target tracking. The example of high-mobility air target tracking demonstrates the feasibility of the method, where two types of the nodes could perform the intelligent selection of the target tracking algorithms. The proposed method provides a solution to the intelligent application of WSNs on target tracking.

Acknowledgments

This work was supported by National Natural Science Foundation (NSF 61071071).

References

[1]. I. F. Akyildiz, et al. "A survey on sensor network," IEEE Communications Magazine, 2002, 40(8):102-114.

[2]. Boella G, Damiano R, Hulstijn J, et al. "A common ontology of agent communication languages: Modeling mental attitudes and social commitments using roles," Applied Ontology, 2007, 2(3-4):217-265.I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[3]. Kaushik N, Chatterjee N. "Automatic relationship extraction from agricultural text for ontology construction," Information Processing in Agriculture, 2017, 5(1): 60-73.

[4]. Maroua Masmoudi, Sana Ben Abdallah Ben Lamine. et al. "An ontology-based monitoring system for multi-source environmental observations ," Procedia Computer Science, 2018(126):1865-1874.A.

[5]. Agnieszka Konys. "Towards Knowledge Handling in Ontology-Based Information Extraction Systems," Procedia Computer Science, 2018(126): 2208-2218.

[6]. Krishnamachari B. "Networking Wireless Sensors," Cambridge University Press, 2005

[7]. K renton J. Malin, "Communicating with Objects: Ontology, Object-Orientations, and the Politics of Communication," Communication Theory, 2016, 26 (3):236-254.

[8]. M.B Alaya, T. Monteil. "FRAMESELF: an ontology- based framework for the self-management of machine-to- machine systems," Concurrency and Computation: Practice and Experience, 2015, 27 (6): 1412-1426.

[9]. Yick, Jennifer, Mukherjee, Biswanath, Ghosal, Dipak. “Wireless sensor network survey,” Computer Networks, 2008, 52(12):2292-2330.

[10]. Lu L, Wang J. “Research on target tracking capture rate calculation and target optical characteristic in target optical imaging tracking system,” Microwave & Optical Technology Letters, 2016, 58(11):2672-2677.

[11]. Xu H, Shen Y. “Target tracking control of mobile robot in diversified manoeuvre modes with a low cost embedded vision system,”. Industrial Robot, 2013, 40(3):275-287.

[12]. Chen S, Huang W. “Maneuvering target tracking from nautical radar images using particle-Kalman filters,” Journal of Electromagnetic Waves & Applications, 2013, 27(18):2366-2378.

[13]. Shenker S, Ratnasamy S, Karp B, et al. “Data-Centric Storage in Sensornets,” Proc. Workshop on Sensor Networks and Applications. 2002:137—142.

[14]. Bouhafs F, Merabti M, Mokhtar H. “A semantic clustering routing protocol for wireless sensor networks,” IEEE Consumer Communications and Networking Conference. IEEE, 2006:351-355.