

ITC 2/47Journal of Information Technology
and Control

Vol. 47 / No. 2 / 2018

pp. 167-183

DOI 10.5755/j01.itc.47.2.19290

© Kaunas University of Technology

**Semantic Data Gathering of Physical Entities in Semantic Sensor
Networks Using Software Agents**

Received 2016/11/20

Accepted after revision 2018/03/07

<http://dx.doi.org/10.5755/j01.itc.47.2.19290>

Semantic Data Gathering of Physical Entities in Semantic Sensor Networks Using Software Agents

Mohammad Ahmadiania

Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, Iran

Ali Movaghar

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Amir Masoud Rahmani

Computer Science, University of Human Development, Sulaimanyah, Iraq

Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, Iran

Corresponding author: ahmadiania@iauk.ac.ir

In a wireless sensor network, with numerous sensor nodes, a huge volume of data is produced. For the massive data generated by sensor networks to be understood by the machine, semantic web technologies such as ontology need to be applied. On the other hand, since the end users intend to access the high-level physical entities information monitored by the sensor network, the applicability of sensor networks can be enhanced through proposing a strategy to extract the data based on entities instead of extracting the raw sensor data. Hence, in this paper firstly, ontology will be presented for physical entities semantic modelling (PESM). Secondly, an appropriate strategy will be offered to collect and aggregate data on physical entities through software agents. The result of modelling and simulation of the proposed method represents its desirable performance to other previous strategies.

KEYWORDS: semantic sensor network, semantic data aggregation, software agents, physical entity modelling, ontology.

1. Introduction

Wireless sensor networks are employed to collect information in areas where the user cannot be present. In such networks, there are numerous sensor nodes spread across the environment, where each node sends

the data from the surrounding environment through other nodes as a stream of data to the sink node. For the massive data generated by sensor networks to be machine-understandable, semantic web technologies

need to be applied. The semantic web technologies can better manage the query and integrate sensor data. They allow the users to perform at a higher level than technical details concerning sensors and sensor networks and perform at domain concepts level. Moreover, they can provide an opportunity to connect sensor data to data available on the web. Ontology is the most important semantic technology to provide and share knowledge. So far, several ontologies for example, CSIRO [17], OntoSensor [12] and SSN [10] have been presented for the semantic modelling of wireless sensor networks and sensor data. Even though there have been many efforts made to collect and aggregate data across a sensor network, the applicability of sensor networks can be further enhanced. It can be done through proposing a strategy on extracting the data semantically and based on entities instead of extracting the raw sensor data, because the end users intend to access the high-level physical entities information that is monitored by the sensor network.

For example, consider the sensor network that is established in a zoo for monitoring of situation of animals and their location. Instead of taking raw data individually, entity-based and interrelated information can be considered. In this example, animals and cages are considered as physical entities. Sensed data gather and store based on these entities (e.g., temperature, humidity, light, water and food amount, etc. from the cages and location, weight, mobility, etc. related to animals). Relation information between the cages, animals-cages and animals are also determined and stored. Therefore, it is possible to answer high level and complex queries.

In entity-based data extraction from the sensor network, an appropriate strategy should be presented to extract the mobile entities information. To collect the entity information and detect the mobile entities, application of software agents can be useful. The software agent is a computer program deployed in an environment that is capable of operating automatically so as to attain its pre-determined objectives [22]. The environment needs to fulfill some conditions prior to the deployment of an agent-based system. According to [22], the environment should be open, dynamic, ambiguous and complex. A wireless sensor network with such conditions can, therefore, be favorable for employing the agents to resolve some problems raised in such networks.

Considering all the facts mentioned above, this paper firstly intends to use a method for semantic and tempo-

ral modelling of physical entities that is monitored by a sensor network and their interrelationships. For this purpose, ontology is used for semantic modelling of real-world physical entities (PESM ontology) [2]. Afterwards, the developments required to model time, space and entity interrelationship will be added to the model. Secondly, an appropriate strategy will be offered to collect and aggregate data on physical entities. To this end, a number of different software agents will be introduced. The next stage involves gathering and aggregation through software agents for physical entities information based on the proposed semantic model.

The rest of the paper has been organized as follows. In Section 2, the previous relevant works will be reviewed. Section 3 will elaborate on the ontology concerning the modelling of physical entities. In Section 4, there is a proposed method for the collection and aggregation of semantic information about physical entities using the software agents. Section 5 will evaluate the proposed solutions. In Section 6, the summary and conclusion of the work are presented.

2. Related Works

In the field of semantic modelling of sensor networks and the use of semantic modelling for the gathering of sensor data, some works have been performed. The majority of works on semantic modelling of WSNs are concerned with appropriate ontology design. In general, the ontologies presented in this regard can be divided into two main categories: sensor-centric ontology and observation-centric ontology. The former has been developed to describe sensors and sensor networks such as CSIRO [17], OntoSensor [12], CESN [9] and SDO [11]. The latter has been developed to describe sensor observations and sensor-obtained data such as Stimuli-Centered [20], O & M [21] and OOSTethys [5]. However, ontology SSN [10] is an appropriate perfect ontology based on OWL2 developed by the W3C Semantic Sensor Network called SSN-XG. This ontology describes the capabilities and properties of sensors, the act of sensing and the resulting observations. Roda and Musulin presented an ontology-based framework based on two ontologies including SSN and SWRL and a new ontology as TAO for the smart management of data collected from the sensor network [19]. In [8], an ontology-based mod-

el is presented to gain access to sensor network data and querying on data stream resources. In this work, the authors allow users to express their needs on a conceptual level independent from language-related details and implementation via developing SPARQL and rewriting the query. Kim et al. presented a methodology to manage the data collected from heterogeneous sensor networks [17]. They introduced a set of words with rules related to the network to connect various networks and manage them using OWL language. Concepts concerned with communications are not considered in SSN ontology and other ontologies of the sensor network. Hence, in [4], an extension was done on SSN ontology which modeled data transmission between sensors and relations between sensors. The aim was to enhance the network lifetime using this model and adapting sensor relations to their area. Calbimonte et al. proposed a new method of characterizing and extracting semantic metadata via analyzing the observations of sensor raw data [7]. Ibrahim et al. presented a temporal-spatial model for sensor networks [14]. The model makes it possible to reason based on sensors in any time and place. In this work, BFO and SSN ontologies are used. Gorrepati and Kim proposed a hierarchical architecture for sensor information description using sensor data so that they become meaningful [13]. Pinheiro et al. presented the Human-Aware Sensor Network Ontology that is a comprehensive alignment and integration of sensing infrastructure ontology and a provenance ontology [18]. Bui et al. presented the analysis, design, implementation, and evaluation of a semantic layer within an existing BASN platform for the purpose of improving the semantic interoperability among sensor networks and applications [6]. Calbimonte et al. proposed an ontology-based approach for providing data access and query capabilities to streaming data sources, allowing users to express their needs at a conceptual level, independent of implementation and language-specific details [8].

3. Physical Entities Semantic Modelling (PESM)

In this section, it is attempted to model the physical entities whose information is collected by the sensor network at a higher level of sensors and their obser-

vation [2]. For this purpose, ontology is presented for semantic modelling of real-world physical entities. DUL upper-level ontology was applied to create the desired ontology. The information relevant to the physical entity can be examined from three aspects: thematic, temporal and spatial information. In designing the ontology of physical entities, it is crucial to take into account the various information aspects of modelling as described in the following.

3.1. Thematic Modelling

In the DUL, the physical entities are modeled as classes derived directly or indirectly from the *DUL:PhysicalObject*. Two entity classes that include *StationaryEntity* and *MobileEntity*, are defined as subclasses of the physical entity class. Each entity has a set of properties, for the definition of which, the *Property* class is applied as a subclass of *DUL:Quality*. Since the properties of the entities can be either constant or vary over time, the two classes *StaticProperty* and *DynamicProperty* are derived from *Property*. Two classes, *ContinuousProperty* and *DiscretProperty*, model continuous properties and discret properties. *DUL:Amount* class was used to model the amount of property. It consists of two parts, including *DataValue* class and *DUL:UniteOfMeasure* class. The parameter *HasProperty* is used to connect an entity to property, while *hasValue* is used to connect the entity to its value.

3.2. Spatial Modelling

DUL:SpaceRegion class of DUL ontology is used to model the location of the physical entity. The geographic location of each physical entity can be determined through a form of polygon or circle. If the geographic location of an entity is an irregular curve, it can be considered in a rough approximation as a polygon with several vertices.

Hence, we have the two classes, *CircleRegion* and *PolygonRegion*, defined as subclasses of *DUL:SpaceRegion*. The *wgs84:point* class in *wgs84* spatial ontology that includes the geographical coordinates of a point is used for defining these classes. As such, the class *PolygonRegion* is defined as a list of points which are actually the polygon vertices (using *rdf:list*), while class *CircleRegion* is defined by a point as center and a radius. At this stage, an entity class can use the parameter *DUL:hasLocation* to connect to one

of the classes derived from the *DUL:SpaceRegion* so as to determine its geographic location.

3.3. Temporal Modelling

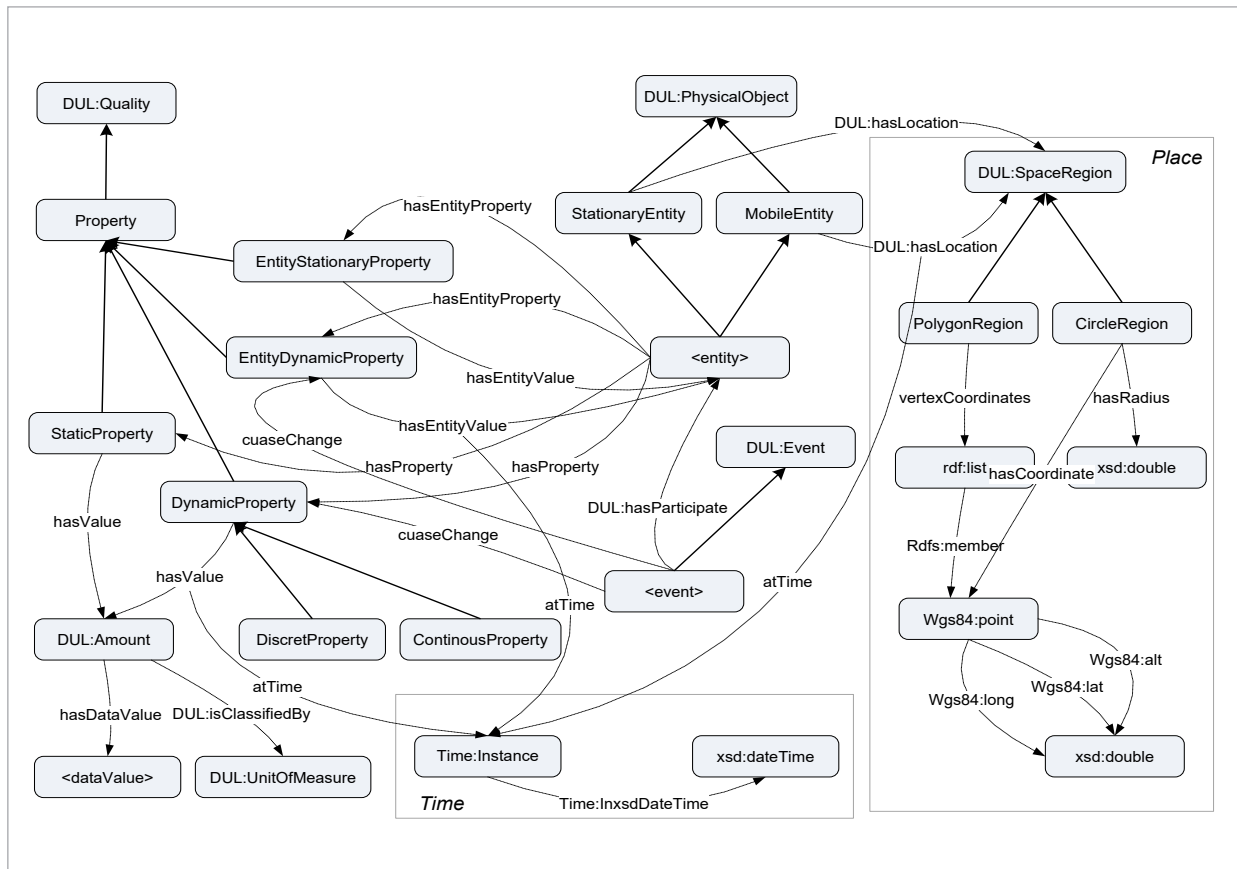
The state of physical entities can be modified from two aspects. First, it can be modified in terms of geographical location, where the location of the mobile physical geography may change by the time, and in spite of this, the geographical location of the stationary physical entities remains fixed and does not change by the time. Second, it can be modified in terms of the subject characteristics, and the value of dynamic characteristics can be changed. To model thematic and spatial changes in time, (*<MobileEntity>*, *DUL:hasLocation*, *<DUL:SpaceRegion>*) and (*<DynamicProperty>*, *hasValue*, *<DUL:Amount>*) triples are considered as dynamic and temporal. In addition, *Time:instance*

class of Time ontology is used to display time. Then, respective triples are related to *Time:instance* class using *atTime* parameter.

3.4. Entity Interrelationships

The various relationships between physical entities can be modeled through the entity properties, which can be either dynamic or static. In order to model the entity static properties, *EntityStationaryProperty* is defined, while the *EntityDynamicProperty* is defined to model the entity dynamic properties, which are linked with *hasEntityProperty* and *hasEntityValue* parameters with entity classes. Because the value of entity dynamic properties can vary over time, the triple (*<EntityDynamicProperty>*, *hasEntityValue*, *<entity>*) is converted into the temporal triple which can vary over time. The mentioned ontology is illustrated in Figure 1 [2].

Figure 1
Physical entities semantic modelling (PESM ontology)



3.5. Semantic Queries

An approach for extracting semantic information of entities from the data obtained based on the semantic model is presented further.

Extracting physical entities information is intended for extracting the entire or part of the semantic information of the entity from the information obtained from the sensor network. The obtained information is determined by the ontology and in terms of RDF triples. SPARQL language is the standard one for semantic query on RDF data. Semantic information preserved with regards to physical entities consists of a large number of thematic, temporal, and spatial information. According to the semantic modeling provided in the previous sections, the framework of semantic information of the entities is partially very similar. Consequently, we present an upper-level pattern to perform queries relating to an entity in a specified and simpler form. Indeed, we intend to provide an entity-based query language at an upper-level. In spite of this, the query provided in this specific form is also converted into SPARQL queries. However, the presentation among the query language increases the precision and simplicity of employing the semantic

data of entities obtained by the sensor network. This query language is an appropriate programming mediator of using the semantic data of the sensors.

The query pattern provided for extracting the information of the entity is illustrated in Figure 2.

According to Figure 2, in this particular pattern firstly, the essential sections of the entity are well defined. These sections involve numerous aspects, as well as time and place. Within this section, we consider the essential parts of the entity semantic information. If the entity semantic information is entirely demanded, sign “*” is defined in this section. Therefore, we select a sensor network on whose data we intend to employ the query. However, once all the data are relevant to one sensor network, this section is no longer essential. In the section *where*, thematic, spatial, and also temporal conditions of the entities under query are well defined. Initially, we consider those features of the physical entity having specific amounts or those that we want to maintain at a specific interval. We use relational operators {<, >, <=, >=, <>} to mention the values as well as the range relevant to the entity features. In this section, virtually, the static features of the entity can be used as identifying an entity.

Figure 2

Query pattern of entities semantic information

```

Select <req_sPropertyI>, ... | <req_dPropertyI>, ... | Time | Loc | *
From [wsn]
Where {
    <det_sPropertyI> <Rel> <sValueI>, ...
    <det_dPropertyI> <Rel> <dValueI>, ...
    Location <Rel_loc> { P, [(x1,y1), ... ,(xn,yn)] | C, [(x1,y1), <R>] }
    Time <Rel_time> { [From : <T>] [To : <T>] | [Instance] }
}
<Rel> ::= { <, >, <=, >=, <> }
<Rel_loc> ::= { EQ, PO, EC, DC, NTPPi, NTP, TTPi, TPP }
<Rel_time> ::= { After, Before, StartedBy, FinishedBy, During, Equals }
<T> ::= <timestamp> | Now | (Y=<y>, M=<m>, D=<d>, H=<h>, I=<i>,
S=<s>)

```

Consequently, the spatial conditions of the entity are specifying after the term Location. that is definite, an address is defined. This address is generally in form of a polygon. Instantly, the letter P is used and the vertices of the polygon are identified with a listing of points. The address is generally specified in a circular form. Then, the letter C is commonly used, and after that the center and radius of the circle are fixed. We should identify the types of relation between specific address along with address of the entity under query applying operators {EQ, PO, EC, DC, NTTPi, NTTP, TPPI, TPP} adopted from [3]. At the end, we determine the temporal conditions. The amounts of the dynamic features of entities relevant to the times set are extracted. To accomplish that, we determine the time. This time is generally a time interval which we determine initially and also end of the interval by *From* and *To*; or simply a time that we identify instantly. To accomplish that, we employ Instance. We employ to fix a specified time. This can be a timestamp, or even a set in terms of the year, month, day, hour, minute, and also second by utilizing signs {Y, M, D, H, I, S}. To associate the identified time to desired times, we utilize temporal relational operators {After, Before, Started, Finished, During, Equals} adopted from [3]. According to the conditions determined in section “where”, it becomes possible to extract one or more physical entities for the results of a query. As was mentioned, queries identified in an upper-level query language are required to be converted to SPARQL language to obtain the appropriate information from the semantic data of the network.

4. Data Gathering of Semantic Entities Using Software Agents

This section provides the proposition of a suitable sensor network strategy for gathering the semantic information of physical entities based on the presented ontology. To gather data from entities by the sensor network, the software agents are employed. In the following, the software agents are introduced along with the software agents used in this work.

4.1. Software Agents Used for Data Gathering

A software agent is a combination of computer software and data, which can operate automatically with-

out the intervention of human agent in a dynamic environment. Some of the software agents are specifically capable of migrating between computers, which are called mobile software agents. The intention is to gather entities’ semantic information on sensor network using software agents. To this end, three different agents are defined and used:

- 1 Sensor agent: These software agents are deployed on sensor nodes.
- 2 Entity agent: An entity agent is created for each entity in the sensor network.
- 3 Sink agent: A sink agent is deployed on the sink node. Later on, the functions of each proposed agent will be discussed.

4.1.1. The Sensor Agent

On each sensor node, there is a static sensor agent. The sensor agent places a semantic layer on the sensor node so as to semantically define and provide the received data. This agent employs the SSN ontology for modelling the sensor, its output, observations, and interconnections. The observation time connects through *ssn: observationResultTime* to a temporal class in the ontology OWL-time, i.e. *instance*. For modelling the location of the sensor node, the *ssn: platform*, class, and parameters of *ssn: onPlatform* and *hasCoordinate* are applied. While the *point* class in ontology wgs84 is used to determine the spatial coordinates. In target tracking applications, this agent detects the target and notifies the sink agent. Moreover, this agent is applied to determine the relationship between the entities that will be discussed in the next sections.

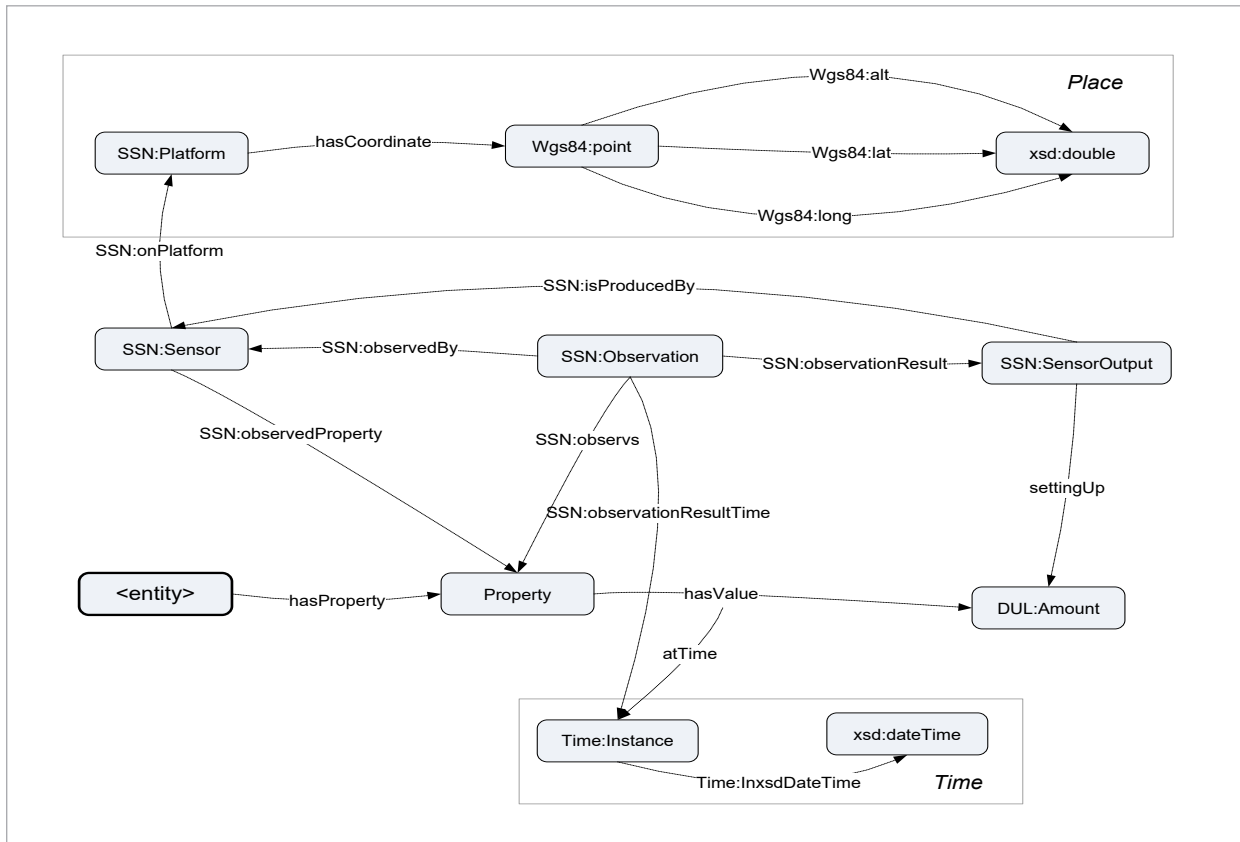
4.1.2. The Entity Agent

The entity agent is used to gather information about the physical entity. It is the task of the agent to migrate through the sensors toward the target area and deploy at a suitable node (leader node) adjacent to the desired entity and then acquire the entity information from sensor agents, which takes place according to the matching of SSN ontology and entity ontology.

For this purpose, as can be seen in Figure 3, the sensor observations are connected to entity ontology through parameters *ssn: observes* and *ssn: observed-Property*, classes *ssn: observation* and *ssn: sensor*, connecting to the entity attributes in the entity ontology, i.e. the *property* class. As such, it is determined what properties a sensor are monitoring the entity. Addi-

Figure 3

Linkaging physical entities ontology to SSN, wgs84 and time ontologies



tionally, the *ssn:sensorOutput* class connects to *DUL:Amount* class using *setting* which indicates the values of an entity attribute. Therefore, the corresponding attributes to a given entity in ontology are evaluated through receiving data from the environment and evaluating the sensor ontology parameters. In order to obtain the time at which an entity changes status or an event occurs, a map is drawn directly between the temporal properties of sensor agent and the entity agent. An observation time in SSN ontology through *ssn:observationResultTime* is linked to the temporal class *instance* in Time ontology. This time can be equivalent to the time when the value of an entity attribute varies. Hence, the valuation time of a property, i.e. the time for *hasValue*, is connected to the same *instance* class.

In order to obtain the spatial information of each entity, we assume that each node knows its location. The distance to the target entity can be estimated through

the calculation of time or strength of the reciprocating signal. In this case, the calculation of target entity location requires the information of at least three sensor nodes surrounding the entity. Whenever an entity agent receives various entity data (temporal, spatial and thematic), the data are sent to the sink agent at the sink node within specified intervals or when the status varies and there is variation in spatial or thematic information. Additionally, the entity agent might receive specific instructions from the sink agent concerning the dispatching of current information or send the entity information in the future. In this case, the entity agent is required to collect and dispatch the entity information as instructed.

4.1.3. The Sink Agent

A sink agent is deployed on the sink node. If the physical entity is detected by any of the sensor agents in the environment or in a pre-defined form, the sink agent will create an entity agent and send it to the

geographic location of the target entity. After that, the task of the sink agent is to make contact with the entity agents to gather information on the entity and complete the entity semantic tree. Having received the information, the sink agent will store the entity information in the semantic and temporal database of the sensor network.

However, the software agents are in charge of certain tasks on issues such as the election of the leader node, tracking of the entity, and detection of the entity interrelationships which will be discussed later.

4.2. Deployment of Entity Agent in an Appropriate Sensor Node

In general, an entity might be placed within the sensing range of several sensor nodes. Thus, multiple nodes can monitor the physical entity and collect its information. Therefore, the entity agent can be deployed on any of these nodes and gather entity information from the entity monitor nodes. However, for gathering data properly to save energy in the network and thereby extending the network lifetime, it is better if the entity agent is placed in a node under certain conditions. For instance, the remaining energy of the node should be higher, the distance between the node and the target should be shorter, the average distance between the node and other monitor nodes of entity should be small, node might entail multiple sensors, and the distance between the leader node and the sink should be shorter.

4.2.1. Entity Detection Algorithm and Leader Node Election

If a physical entity is placed in the sensor network, the sensor nodes hosting the entity within their sensing range will detect that particular entity. These nodes are called monitor nodes over that entity. The task of sensor agents of monitor nodes is to elect one of the monitor nodes as the leader. The procedure is as follows: each sensor agent calculates the competency of the corresponding node for being a leader according to predefined parameters, notifying the other sensor agents in the monitor nodes so as to elect the node with the highest level of competence as the leader node. In order to calculate the level of competency (C), each parameter for election of a leader is calculated and normalized as can be seen in the following.

The remaining energy (e_k): In order to calculate the

remaining energy parameter of each node, the remaining energy node (e_i) is divided by the maximum energy of nodes e_{max} :

$$e_k = \frac{e_i}{e_{max}}. \quad (1)$$

The distance from the target entity (d_k): In order to calculate this parameter, the distance between the node and target entity (d_i) is divided by the sensing range of node (r_s), then its supplement relation to one will be obtained. According to the following equation:

$$d_k = 1 - \frac{d_i}{r_s}. \quad (2)$$

In cases where the sensor node is located on the entity, e.g. if the target entity is a geographic area, the distance to the target can be considered zero, or if adjacency to the center is significant, then the node-to-center distance will be considered.

Multi-sensing (g_k): In order to calculate the multi-sensing parameter, the total number of entity properties achievable by the node (or the number of different sensors of a node) plus one (in case there is a positioning property in the nodes) is considered as the number of node sensings (g_i) divided by the total number of sensible entity properties in the network (g_{total}) plus one (indicating the location):

$$g_k = \frac{g_i}{g_{total}+1}. \quad (3)$$

The distance from the sink node (s_k): In order to calculate this parameter, the distance between the monitor node and the sink node or the number of steps to the sink node (s_i) is divided by the largest diameter of the sensor network or the maximum number of steps in the network (s_{max}):

$$s_k = \frac{s_i}{s_{max}}. \quad (4)$$

The competency of a node for becoming the leader is considered as an outcome of the mentioned parameters. The effect of various parameters on the selection of leader node, however, can vary in different applications, where different parameters do not leave an identical impact. For example, in most applications, the two parameters d_k and e_k are more important. In order to adjust the effect of parameters in calculating

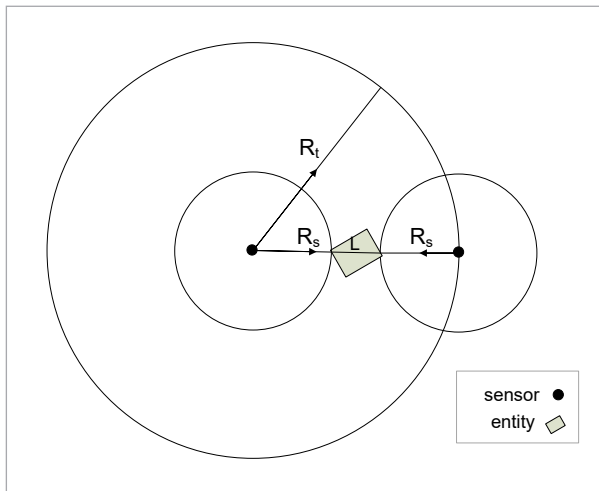
the competency, each parameter is assigned a value between zero and one so that the total weight is equal to one:

$$C_k = w_e \times e_k + w_d \times d_k + w_g \times g_k + w_s \times s_k$$

$$w_e + w_d + w_g + w_s = 1. \tag{5}$$

As such, the value C_k represents a number between zero and one, indicating the competency of the k th node to become a leader that determined in each node. Then, each sensor agent broadcasts to its neighbors the calculated value of C parameter. If the size of the largest of entity diameter, i.e. the distance between the remotest points in target entities, is considered equal to L , there will be two scenarios. Firstly, the communication radius of the sensor nodes is twice as large as the sensing radius plus L . i.e. $R_t > 2R_s + L$ (Figure 4). In this case, all the sensor nodes detecting the target are in each other neighborhood. After one broadcast, all the monitor nodes entail the competency value of other nodes as the sensor agents deployed in the monitor nodes can specify the leader node (i.e. the node with the highest C value) through comparison of C values.

Figure 4
Relation between communication radius and sensing radius, $R_t = 2R_s + L$



The second scenario involves $R_t < 2R_s + L$. In this case, all the monitor nodes might not be neighbors. Therefore, after one broadcast, some of the monitor nodes might not have received the C value of some other

monitor nodes. After the first-stage broadcast, each node has a list of neighbors and their C values which are monitor nodes of an entity. The proposed solution to receiving the list of entire monitor nodes by each of the monitor nodes is that each monitor node should send a list of attributes and competencies of its monitor neighbors to all the monitor neighbors. Thus, if two monitor nodes are two hops away from one another, they can receive each other profile and complete their list. This will be repeated as long as the list of the entire monitor nodes is completed. The number of iterations (K) is limited, and is obtained through Equation (6) assuming that the network is full-connected and there is a full environmental coverage:

$$K = \left\lceil \frac{2R_s + L}{R_t} \right\rceil. \tag{6}$$

After each monitor node obtains a list of all entity-monitoring nodes and their competencies, it can select a monitor node with the highest level of competency as a leader node. In addition, it is possible that two or more monitor nodes to be equal in terms of competency. In this case, a monitor node with the lowest ID will be elected as the leader node. After selecting a node as the leader node by the sensor entities, the sensor agent deployed on the leader node will send a request to the sink agent as the sink agent dispatches a desirable entity agent toward the leader node.

4.3. Scheduling of Sensor Agents and Data Gathering

After the deployment of the entity agent on the leader node, at first, receive the characteristics of the sensor agents concerning the monitor nodes in order to determine whether any sensor element will collect the information from the entity. Consider the possibility of geographical location and finally measure the remaining energy and its distance to the target. In order to save energy, the entity agent designs and sends a timetable for activities of the sensor agents. Based on the scheduling, they will focus on the physical entity and send the information to the entity agent.

Since the leader node – on which the entity agent is deployed – is constantly active performing more transferring and processing tasks than other monitor nodes, its energy discharges after a while and dissipates earlier than the rest of the nodes. To prevent this, the entity agent should, after a specified peri-

od, be transferred from the current node to another sensor node. To do so, another monitor node, whose competency is higher than the rest of the monitor nodes according to the list of monitor nodes, should be elected as a leader node to notify the substitution to other nodes. Then, the entity agent is transferred to the new leader as the former leader turns into a regular monitor node. Along with the current entity agent, the schedule of the monitor nodes and the list of node properties will be transferred. The energy in this way will be consumed moderately in the network.

4.4. Tracking of Mobile Entities

An entity whose information has been collected by the entity agent can be mobile and move around the environment like military vehicles moving on the battlefield or an animal roaming in the environment about which we intend to collect information. In this case, as the entity moves in the environment, naturally, nodes surrounding the entity, which can collect information, are changed. Consequently, a series of nodes as entity monitor nodes depart from the physical entity and thus make it impossible to detect and monitor the entity. The sensor agent in these nodes notifies the entity agent in the leader node. The leader node should eliminate these nodes from the list. The nodes whose entity has been approached and laid within their sensing radius can detect the entity. The sensor agent of these nodes becomes activated, calculating the competency of the node for being a leader and then spreading the nodal properties as a broadcast to the neighbors. If the leader node is in the neighborhood of the new monitor node, the properties of the monitor node will be added to the list by the entity agent and the node is added to the timetable, notifying the new monitor node the involvement of the leader node. If the leader node is not in the neighborhood of the new monitor node and no message from the leader node is sent to the monitor node, then the monitor node will resend the message to its neighbors. Each neighboring monitor node can send the information of leader node to the new node and also send the information of the new monitor node to the leader node. If the monitor node still does not receive the message (assuming that the sensor network is connected to a network), it implies that a new entity has engaged in the sensor network and there has not been any entity agent created to collect the new information. In this case, the

algorithm of selecting the leader node is exercised as discussed earlier.

When the target entity is moving, the target entity departs from the leader node, thus curtailing its competency level. Prior to monitoring the entity, the entity agent should migrate to new monitor nodes in the vicinity of the entity with high competencies.

4.5. Gathering Relationship Information Between the Entities

The physical entities can build the thematic and spatial relationship with each other. In this section, the procedure to extract these relationships from the sensor network using the software agents is discussed. The early detection of a relationship between two entities is performed by overlapping sensor nodes i.e. the sensor nodes will recognize two different entities. If the sensor agent in a monitor node is monitoring over an entity for example y , it detects another physical entity, for example x , or is selected as monitor node by another entity agent, then it will introduce entity y to entity agent of x and entity x to the entity agent of y . The sensor agent and even the entity agents cannot determine the type of relationship between entities on their own. Thus, the entity agents will notify the sink agent to deduce the type of relationship between two entities with regard to complete relevant information and then add this information to a semantic tree of both entities.

In addition, if a moving entity is departing from another entity, one of the entities might now leave the sensing range in the sense agents having been so far monitoring over the two entities. In this case, the sensor agent will immediately notify the two relevant entities as the entity agents will in turn, notify the sink node which will examine the type of entity relationship so as to either add or remove information at the entity semantic trees.

5. Evaluating the Proposed Methods

The strategies are investigated from two aspects. Firstly, because the proposed solutions are applied on wireless sensor networks, the efficiency of sensor networks is assessed by applying the proposed methods. For that purpose, there are several param-

eters to be evaluated, such as energy consumption and network lifetime. Secondly, the information extraction accuracy of physical entities will be assessed so as to analyze the proposed method for semantic modelling. In the evaluation, the proposed algorithm is briefly called *PESM*.

5.1. Simulation Tools and Conditions

The open source network simulator, J-sim, is used to assess the strategies presented. To work with different ontology, Jena library is used.

A protected area, which is the habitat for gazelles, is considered for evaluation of the proposed method. In this area, a sensor network has been set up for monitoring gazelles and their habitat. The objective of setting up a sensor network in this environment was to collect information regarding geographical regions such as climate information and also information about gazelles. This is a proper example of using this ontology because it has different mobile and stationary physical entities and each entity has various static and dynamic properties. Various relationships also exist among entities. Physical entities of this scenario are 1) Geographical region 2) Gazelle. Each geographical region has static properties such as name, region type, vegetation type, and so on. The dynamic properties such as temperature, air pressure, solar radiation amount, humidity, or others. Each gazelle has also static properties such as identification number, color, sex, species, or anything else and dynamic properties such as speed, weight, location, and the rest. Furthermore, some information related to the gazelle such as the climate information of the gazelle environment is the same as the information of the region in which the gazelle is located. The used climate dataset in this work involves climate data obtained from the Meteorology Research Center in Kerman, Iran. These data were collected by ten automatic weather stations based in Kerman province, and twenty gazelles are assumed in the environment.

In this scenario, a sensor network is considered for a given area with dimensions of 1000×1000 m. The sink node is settled in the center of the environment. The sensor nodes are located in the environment randomly. It is assumed that each sensor node is able to send single-hop data to the sink. Other assumptions of the simulation are expressed in Table 1 briefly. Each experiment has been repeated 10 times for each

sensor numbers, and the achieved result is the average of 10 executions.

Table 1

Conditions of simulation

sensor node numbers	20, 50, 100, 150, 200, 300
dimensions of environment	1000 × 1000 m
period of data sending	10 minutes
initial energy of the sensor nodes	in [1.8, 2] Jules
size of data packages	526 bits
size of control packages	8 bits

The assumption is that the dynamic information from the environment might not be identical in all parts of the area. In this case, adjacent points in the network usually have identical information. In this situation, a geographic area can be divided into several geographic areas, where similar information can be extracted. Each of these geographic areas is considered a physical entity. In order to identify the geographic areas and sensors deployed in a geographic area, the clustering method is employed based on environmental similarity and cellular learning automata[1]. After running this algorithm, certain geographic areas are classified into geographic regions. With the implementation of the algorithm, the number of geographic areas, i.e. the applied physical entities, will amount to fifteen geographic areas.

5.2. The Use of PESM Ontology

In this scenario, PESM ontology is combined with SSN, temporal and spatial ontologies. Part of the physical entity modelling of a geographical region is presented in Figure 5. In this figure, only one static property (region title) and one dynamic property (temperature) are considered. In addition, a geographical region is a static entity and its shape is considered as a circle. Figure 6 presents part of the semantic modelling for a gazelle physical entity using the proposed ontology. Due to lack of space, in this figure, too, only one static property (gazelle color) and one dynamic property (moving speed) are presented.

Figure 5
Simplified view of semantic modelling of a geographical region as a stationary physical entity

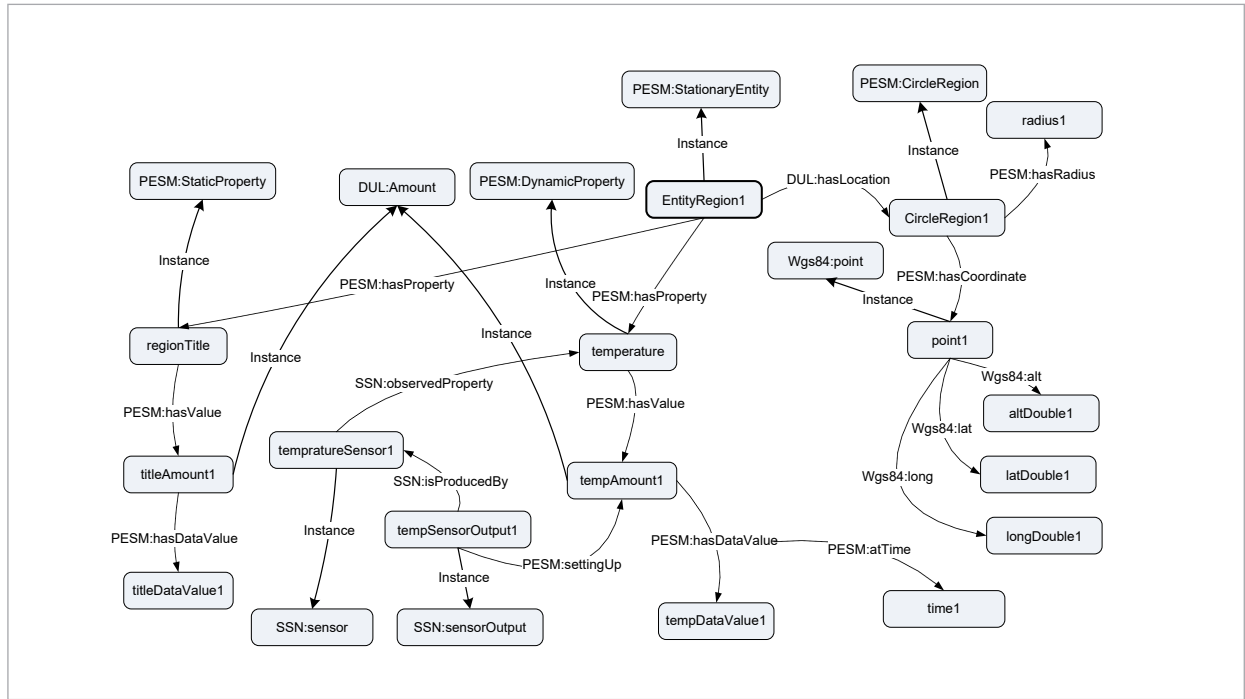
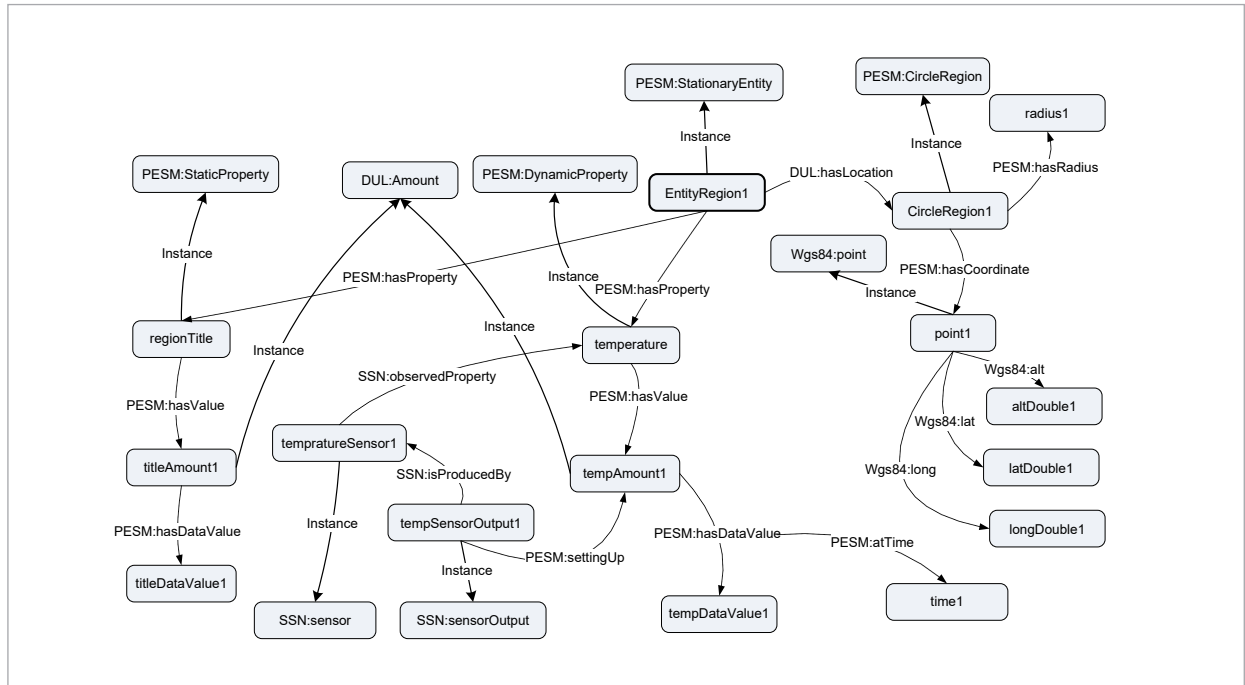


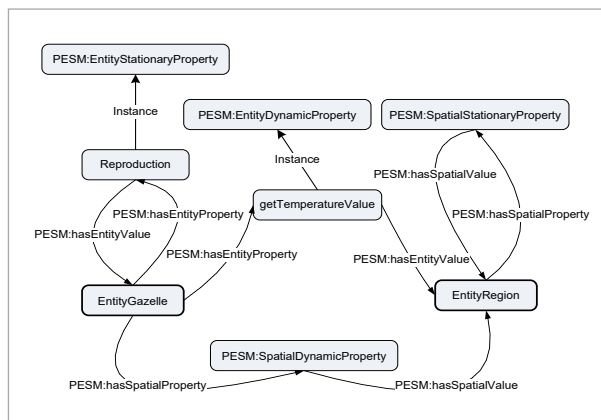
Figure 6
Simplified view of semantic modelling of gazelle as a mobile physical entity



Different entities have relationships with each other such as spatial relationships of geographical regions with each other or gazelle relationships with each other or the relationship of gazelles with geographical regions. In Figure 7, part of the modelling for relations of entities is presented. Geographical region entities can have any of the spatial relationships with each other. Gazelles can have entity relationships with each other, such as reproduction relationship (parent and child). However, gazelles can have any of the spatial relationships or entity relationship with geographical regions. For example, the temperature of gazelle environment is obtained from the region in which the gazelle is located.

Figure 7

Simplified view of semantic modelling of relations between physical entities in example



5.3. Evaluation of Parameters Concerning the Performance of Sensor Networks

It is crucial to assess the efficiency of parameters in the proposed method concerning the performance of sensor networks. A major portion of the solutions presented in this paper focuses on clustering, gathering and aggregating data in a wireless sensor network. Hence, the work needs to be compared to relevant works on clustering and data aggregation in sensor networks. Considering the proposed solution, some of the previous works were picked for the purpose of comparison:

- 1 *GPSR* method [14]: A basic data gathering method in sensor networks where data are gathered without clustering. In this way, data gathering does not

take place semantically. In addition, all sensor network nodes are active.

- 2 *LA-SleepScheduling* method [1]: It is a clustering algorithm based on environmental similarity through cellular automata learning technique. In this work, after clustering, a scheduling algorithm is provided to enable sensor nodes for rotation sensing over the environment. In this way, the sensing data are not modeled semantically.
- 3 The algorithm presented in [7] is briefly called *DSSM-RM*: In this work, the data gathered by sensors have been modeled through ontology. In this method, all sensor nodes are active. Unlike the proposed method, the semantic modelling is based on sensors, not on the entity.

The parameters used are as follows: Network lifetime, the number of packets received by the sink node and algorithm overhead. The experimental results are displayed in Figures 8 to 10.

In the first experiment, the simulation was estimated to last one month, i.e. 43,200 minutes. In the second experiment, the simulation continues until the first node dissipates.

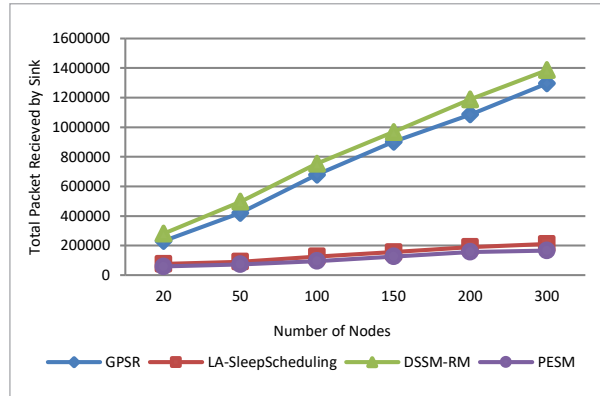
5.3.1. The First Experiment

In the first experiment, the number of packets received by the sink node was compared in different methods. The lower the number of packets sent by the nodes in a network, the lower the amount of energy consumption in the network and the higher the network longevity. Figure 8 indicates the results of evaluations. The proposed method sends the minimum number of packets to the sink node while the *LA-Sleep Scheduling* similarly sends few packets to the sink node. In the proposed method and *LA-Sleep Scheduling*, unlike other methods, the increased number of sensors will not raise the number of sent packets. This is because the higher numbers of nodes keep the number of active nodes roughly constant. In fact, the number of packets sent in these two methods is proportional to the number of entities in the environment, not the number of nodes in a sensor network. The reason why the proposed method sends slightly fewer packets than *LA-Sleep Scheduling* is that the proposed method involves only one sensor node (node entailing an entity agent) sending information packets for each physical entity or geographic area to the sink, inspite of this in *LA-Sleep Scheduling*, all

active nodes send packets to the sink. There are two active nodes here for almost every geographic area.

Figure 8

Comparison of different methods with respect to total received packets in Sink



5.3.2. The Second Experiment

In the second experiment, the network lifetime of different methods is compared together. The network lifetime in this research was considered as the dead time of the first node in the network [14]. The evaluation results are shown in Figure 9, which suggests that the efficiency of the proposed method is desirable. As compared to the *GPSR* and *DSSM-RM*, the proposed method can greatly boost the network lifetime. However, regarding network longevity, it performs not as desirable as *LA-Sleep Scheduling* because *LA-Sleep Scheduling* does not collect information semantically. Despite sending fewer packets, the proposed method carries out more processing tasks, transfers inter-network information concerning for example migration of the entity agent, and sends larger packets to the sink (for having more information on the entity as well as semantic annotations), and finally, it has a shorter lifetime than the *LA-Sleep Scheduling*. Figure 10 comparing the percentage of energy consumption in algorithm in regard to the overall energy per unit of time consumed in the network, i.e. the algorithm expenses in various methods. The ratio in the presented algorithm is higher than that of any other algorithms. Although, this is due to the presented method employs essentially the *LA-SleepScheduling* algorithm. Furthermore, the application of software agents and also semantic data annotations involve specific over-head. Due to the fact

that most of the time, the most of the nodes are inactive, even though, the network lifetime is considerably improved as higher energy consumption in the learning phase is much more appropriate compared to the many other methods shown in Figure 9.

Figure 9

Comparison of different methods with respect to lifetime of network

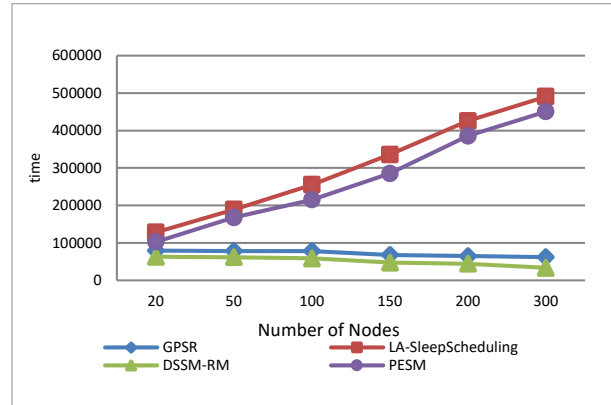
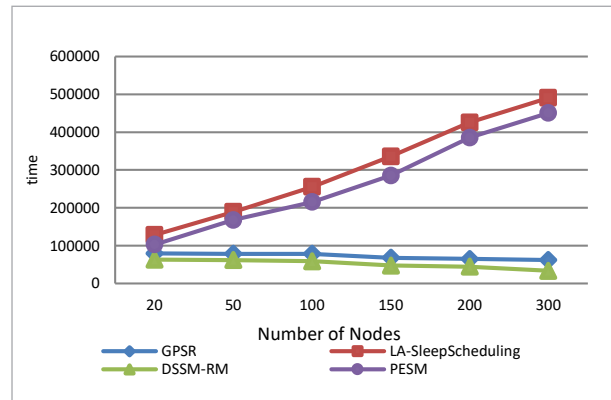


Figure 10

Comparison of different methods with respect to algorithm overhead



5.4. Evaluation of the Extraction of Physical Entities Semantic Information

After gathering the semantic information of entities from the environment, it becomes possible to extract the information of the entities by applying semantic queries.

In this section, we intend to assess the accuracy of extracting the semantic information of entities. In

order to evaluate the accuracy of the information extraction, we compare the proposed method with the DSSM-RM method [7] which collects the information of the sensor network based on sensors and semantically. Parameters under comparison include variables *Precision* and for responding to queries.

Employing entity-based queries to collect individual and relational information of entities is performed dependent on the method provided in this article. However, in *DSSM-RM* which semantic data are obtained depending on sensor, absolutely no direct method is presented to apply entity-based queries. Therefore, to extract the information of an entity depending on *DSSM-RM* method, we extract information obtained by the entirety of the monitor sensors of the entity. In the difficulty under study, monitor sensors are positioned in the respective geographical region. Consequently, to provide the information of an entity in an integrated form, we should provide the mean values of the attributes obtained by the sensors that have collected for each attribute. To deal with the queries concerning the geographical entities, depending on the attributes regarded in the query, the attributes being considered by all of the monitor sensors of geographical entities are employed. Consequently, since it may be accomplished that values relevant to an attribute obtained by all of the various monitor sensors of a geographical entity are all different, a threshold like T (in %) is regarded. At this point, to consider whether an entity is considered as the result of a query or not, the values obtained by the monitor sensor within that entity are examined. Once at least % T of the monitor sensors of the entity accommodates the condition of a query, the entity might be regarded as the result of the query and also its proper information will become returned. Meanwhile, the *DSSM-RM* model does not communicate, modeled, or aggregation the entity information. Accordingly, it is not possible to implement queries concerned with the communication between entities. Hence, to compare the two methods, 20 queries are designed to separately extract the information related to entities.

To assess the precision of responses to queries, two variables *Precision* and *Recall* are used. In the *DSSM-RM* method, selecting T value is effective. That is, increasing T value will enhance *Precision* yet reduce *Recall*. The reduction of T value will reduce *Precision* yet enhance *Recall*. Accordingly, to reach a balance

between the two variables, some tests are carried out using various T values. As a result, if these tests, T value %87 is used in simulation and comparison. Results from comparing two methods with respect to these variables can be seen in Figures 11 and 12.

Figure 11

Comparison of proposed method to DSSM-RM with respect to Recall variable for responding queries

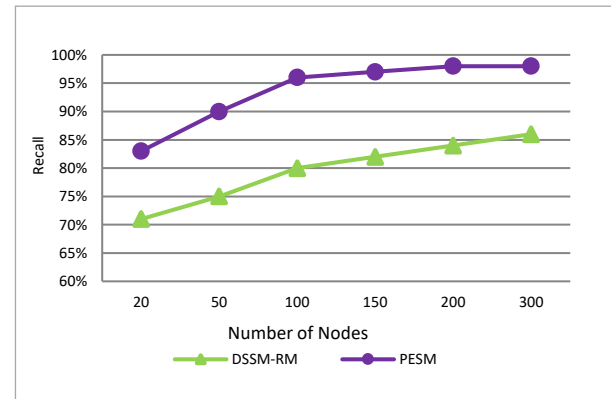
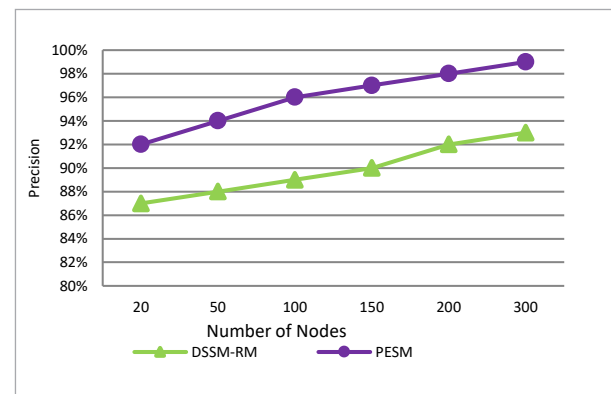


Figure 12

Comparison of proposed method to DSSM-RM with respect to the Precision variable for responding queries



Recall variable approves the ratio of the many appropriate entities extracted to the total number of desired entities for responding a query. The results of comparing both methods with regards to variable *Recall* is shown in Figure 11. Whenever there are certainly a small number of sensor nodes, the random dispersion of the sensor nodes in the environment may cause the absence of the sensor nodes in some regions. As a consequence, respective entity information is not ob-

tained. Therefore, the information of several entities will not exist in the knowledge base and *Recall* value will certainly decrease. Due to the fact that the number of sensor nodes also increases in both of these methods, *Recall* value is significantly higher in the presented method compared with the *DSSM-RM* method. This is because, in the proposed method, the information related to the entity is collected by the right nodes (regarding various factors like the amount of energy remained, distance to the target or centrality and so on).

Whenever, each entity attribute has actually certain information in the knowledge base which might be totally different. Therefore, it can be possible that some appropriate entities are not recovered in entities query.

Precision variable illustrates the number of appropriate entities extracted to the whole number of the entities extracted as compared to the results of the query. Figure 12 compares two methods with respect to *Precision* variable. The proposed method outperforms than the *DSSM-RM* method (regarding variable *Precision*). In sum, the proposed method has performed better (regarding *Precision* variables and *Recall*). The reason that *Precision* variables are not 100 % in the proposed method is that the values that different sensors attained in a geographic area cannot be completely similar. Hence, a small amount (for 2 % of nodes), noise or environmental differences are considered. Those sensors taken as the monitor or repre-

sentative for sensing an entity may sense amounts different from the mean values of a region.

Conclusions

This paper attempted to provide a method which supports both semantic and entity-based to gather data from sensor networks. As a result, the collected data can be understood by the machine and provide high transparency to be applied by users. Hence, this paper firstly presents a method for semantic and temporal modelling of physical entities. To this end, ontology was presented for semantic modelling of real-world physical entities. Afterwards, the developments required to model time, space, and entity relationships were added to the model. Then, an appropriate strategy was offered to gather and aggregate data on physical entities. In fact, it involves the gathering and aggregation through software agents for physical entities information based on the proposed semantic model. In addition, a strategy was presented to track and collect information about the mobile physical entities assisted by software agents. In several experiments, the proposed methods were evaluated with regard to various parameters. The results indicated that the proposed method is adequately efficient as compared to previously developed strategies.

References

1. Ahmadinia, M., Alinejad-Rokny, H., Ahangarikiasari, H. Data Aggregation in Wireless Sensor Networks Based on Environmental Similarity: A Learning Automata Approach. *Journal of Networks*, 2014, 9(10), 2567-2573. <https://doi.org/10.4304/jnw.9.10.2567-2573>
2. Ahmadinia, M., Movaghar, A., Rahmani, A. M. Ontology-Based Modeling and Information Extracting of Physical Entities in Semantic Sensor Networks. *IETE Journal of Research*, 2018, 64(1), 1-17.
3. Batsakis, S., Petrakis, E.G. SOWL: Spatio-Temporal Representation, Reasoning and Querying over the Semantic Web. 6th International Conference on Semantic Systems, Graz, Austria, 2010, 1-9. <https://doi.org/10.1145/1839707.1839726>
4. Bendadouche, R., Roussey, C., De-Sousa G., Chanet, J. P., Hou, K. M. Extension of the Semantic Sensor Network Ontology for Wireless Sensor Networks: The Stimulus-WSNnode-Communication Pattern. In 5th International Workshop on Semantic Sensor Networks in conjunction with the 11th International Semantic Web Conference (ISWC), 2012, 16-p.
5. Bermudez, L. OGC Ocean Science Interoperability Experiment. Phase II Report. OGC Engineering Report Open Geospatial Consortium, 2010.
6. Bui, V., Brandt, P., Liu, H., Basten, T., Lukkien, J. Semantic Interoperability in Body Area Sensor Networks and Applications. In Proceedings of the 9th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, 29, 210-216. <https://doi.org/10.4108/icst.bodynets.2014.257042>
7. Calbimonte, J. P., Corcho, O., Yan, Z., Jeung, H., Aberer,

- K. Deriving Semantic Sensor Metadata from Raw Measurements. 5th International Workshop on Semantic Sensor Networks, in conjunction with the 11th International Semantic Web Conference (ISWC), 2012.
8. Calbimonte, J. P., Jeung, H. Y., Corcho, O., Aberer, K. Enabling Query Technologies for the Semantic Sensor Web. *International Journal on Semantic Web and Information Systems*, 2012, 8(EPFL-ARTICLE-183971), 43-63.
 9. Calder, M., Morris, R. A., Peri, F. Machine Reasoning About Anomalous Sensor Data. *Ecological Informatics*, 2010, 5(1), 9-18. <https://doi.org/10.1016/j.ecoinf.2009.08.007>
 10. Compton, M., Barnaghi, P., Bermudez L., García-Castro R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V. The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2012, 17, 25-32. <https://doi.org/10.1016/j.websem.2012.05.003>
 11. Eid, M., Liscano, R., El-Saddik, A. A Universal Ontology for Sensor Networks Data. In *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSAS)*, 2007, 59-62. <https://doi.org/10.1109/CIMSAS.2007.4362539>
 12. Goodwin, J. C., Russomanno, D. J., Qualls, J. Survey of Semantic Extensions to UDDI: Implications for Sensor Services. In *SWWS*, 2007, 16-22.
 13. Gorrepati, R. R., Kim, D. H. A Hierarchical Architecture for Semantic Representation of Sensing Information in Pig Farm. *International Journal of Control and Automation*, 2014, 7(7), 55-64. <https://doi.org/10.14257/ijca.2014.7.7.05>
 14. Ibrahim, A., Carrez, F., Moessner, K. Spatio-Temporal Model for Role Assignment in Wireless Sensor Networks. In *Proceedings of the 19th European Wireless Conference (EW)*, 2013, 16, 1-6.
 15. Karp, B., Kung, H. T. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ACM, 2000, 243-254. <https://doi.org/10.1145/345910.345953>
 16. Kim, D. Y., Cha, S. H., Cho, K. H. Ontology-Based Methodology for Managing Heterogeneous Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, 2013, 9(9), 610684.
 17. Neuhaush, C. The Semantic Sensor Network Ontology: A Generic Language to Describe Sensor Assets. In *12th AGILE International Conference on Geographic Information Science, Workshop on Challenges in Geospatial Data Harmonisation*, Hannover, Germany, 2009.
 18. Pinheiro, P., McGuinness, D. L., Santos, H. Human-Aware Sensor Network Ontology: Semantic Support for Empirical Data Collection. In *Proceedings of the 5th Workshop on Linked Science*, Bethlehem, PA, USA, 2015.
 19. Roda, F., Musulin, E. An Ontology-Based Framework to Support Intelligent Data Analysis of Sensor Measurements. *Expert Systems with Applications*, 2014, 41(17), 7914-7926. <https://doi.org/10.1016/j.eswa.2014.06.033>
 20. Stasch, C., Janowicz, K., Bröring, A., Reis, I., Kuhn, W. A Stimulus-Centric Algebraic Approach to Sensors and Observations. In *GeoSensor Networks*, Springer Berlin Heidelberg, 2009, 1, 169-179. https://doi.org/10.1007/978-3-642-02903-5_17
 21. Wei, W., Barnaghi, P. Semantic Annotation and Reasoning for Sensor Data. In *Smart Sensing and Context*, Springer, Berlin Heidelberg, 2009, 66-76.
 22. Weiss, G. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT press, 1999.