

# Conceptual metadata model for sensor data abstraction in IoT environments

Yoosang Park, Jongsun Choi and Jaeyoung Choi\*

School of Computer Science and Engineering, Soongsil University, Seoul 06978  
KOREA

\*E-mail: [choi@ssu.ac.kr](mailto:choi@ssu.ac.kr)

**Abstract.** Sensor data abstraction is necessary to provide users with context-aware services. Sensor data abstraction mechanism in context-aware system usually consists of collecting data, converting, and context reasoning. For this mechanism in IoT environments, sensor data is used that is described in the pairs of key-value set or digit value. However, environmental data cannot be sufficiently formed since these sensors data are provided with these values. For example, it may need metadata of contexts and things description. In this paper, we propose a new conceptual metadata model for sensor data abstraction in IoT environments. The proposed model provides sensor data and their metadata as low-level context, which is a part of basic context for presenting given environment, to context-aware system. In the experiments, we describe a procedure to generate low-level context for sensor data abstraction based on the proposed model and to provide this information to the context-aware system.

## 1. Introduction

Context-aware system provides users with context-aware services based on context information. At this point, context refers to evaluated state information that applies current circumstances, users, and their surroundings. One of the straightforward examples is described as follows: "a certain sensor device sensed temperature as warm state."

This context can be presented from sensor data processing. Procedures for processing sensor data normally consist of three major steps: collecting sensor data, converting this collected data into state information, and reasoning this state information for application services. These steps are known as sensor data abstraction [1,2]. The main purpose of sensor data abstraction is to provide understandable content to users, and related systems as appropriate data form, which is context. Sensor data abstraction helps to accumulate sensor data with any relevant piece of meaningful information. However, this sensor data abstraction cannot be done by individual sensor data. Because the users and systems have no hints for choosing sensor data state, and they need guidelines for making decisions of this sensor data. Sensor devices should be given more information to deal with these problems, for example, profile, device specification, their characteristics. And they are generally called as metadata. However, sensor devices in IoT environments do not provide these metadata with sensor data, and it contains key-value based dataset or sensed digit values. In these circumstances, metadata model for converting sensor data into state information is necessary. In this paper, we propose a new conceptual metadata model for sensor data abstraction in IoT environments. The proposed model consists of two major data fields, which are STATICMETADATA and ACTIVEMETADATA.



STATICMETADATA carries fixed data, for example, profile, device specification, and device characteristics. And ACTIVEMETADATA carries changeable values such as access information, entity name, types of sensed values, and actual sensor data values. Through the proposed model, sensor data can be formalized for sensor data abstraction with detail contents.

## 2. Related work

The main concept of context was mentioned by the research of Schilit and Theimer [3], and later Abowd and Dey [4] introduced context-awareness. Until now, they are still important ideas for context related researches. Perera et al. [5] and Maarala et al. [6] represented and outlined processing models and systems for sensor data abstraction. And they are organized by three major procedures: sensor data acquisition, converting sensor data into state information, and reasoning this state information for application services. Bettini et al. [7] summarized data abstraction levels as three degrees with their individual purposes and representation forms as low-level context information, high-level context, and situational relationships.

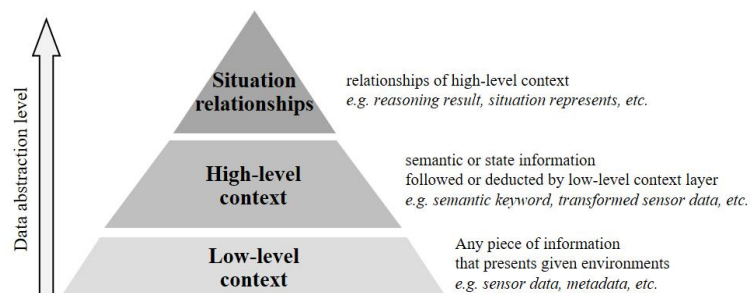
Meanwhile, for systems to process these contexts, researches for representing context are mainly conducted with three major representing forms as key-value based, mark-up scheme based, and ontology-based. Firstly, key-value based context representation is mentioned by a research of Nambi et al. [8] and their main purpose was designing knowledge base. Sensor data acquisition in this research to build the knowledge base, is done by collecting data by key-value forms. Key-value based representation appears to be a pair of an identifier and sensed value. They considered sensed value as raw data. Secondly, Mark-up scheme based context representation is mentioned by a research of Knappmeyer et al. [9] and its main aim was to transmit context to remote sites. For this situation, ContextML was suggested. Mark-up scheme based context representation holds information such as profile and sensed values in a structured way. Finally, ontology-based context representation is mentioned by a research of Kayes et al. [10] and their main aim was to make decisions for each patient in health facilities. Ontology-based context representation is able to describe RDF-based entities as an identifier with each patient state and they are considered individually. RDF format supports universal description methods for those who describe any documents representing states.

By the researches above we followed, key-value based context representation which contains the identifier and sensed value and is the simplest representing way. However, it is difficult to contain additional information which can lead to reasoning techniques because ontology procedure needs much more information than key-value based context representation that only contains the identifier and raw data in a restricted format.

## 3. Proposed metadata model

### 3.1. Context levels in sensor data abstraction

Context levels in sensor data abstraction are arranged as three levels: low-level context, high-level context, and situational relationships followed by the research of Bettini et al. [7] in section 2. Figure 1 shows that context levels in sensor data abstraction and outlines for each level. Details for each level description are as follows.



**Figure 1.** Context level according to the data abstraction.

### *3.1.1. Low-level context*

Low-level context is the bottom level of the sensor data abstraction and it contains any part of information that represents given environments, for example, sensor data, metadata, etc. Sensor data is sensed value measured by a certain sensor device and represents key-value set or digit values. Metadata refers to data about data, and any object information in real-world, such as profile, device specification, etc. Metadata is typically described in a structured format such as XML or JSON. There are two main input data in low-level context; sensor data and metadata. Output data is a formed entity that is considered input data in high-level context for the next step.

### *3.1.2. High-level context*

High-level context level carries semantic information or state information. In this level, output data from the low-level context is considered as input data, and then they are processed to create state information based on the structured sensor data converting scheme. Output data is formed as entity which contains each identifier in the real world and converted sensor data into state information. State information is described as keywords which are able to be accessed by context-aware system.

### *3.1.3. Situation relationships*

Situation relationships level carries the top-notch context in this paper, and it is directly connected to services in real-world applications. To provide these services, entities at this level are formed as triplets of {ENTITY, PREDICATE, ENTITY}. With this representation, the context indicates certain entities and a relationship between them, so that it is able to express surroundings for each state information processed by the guideline and its condition is defined by the users.

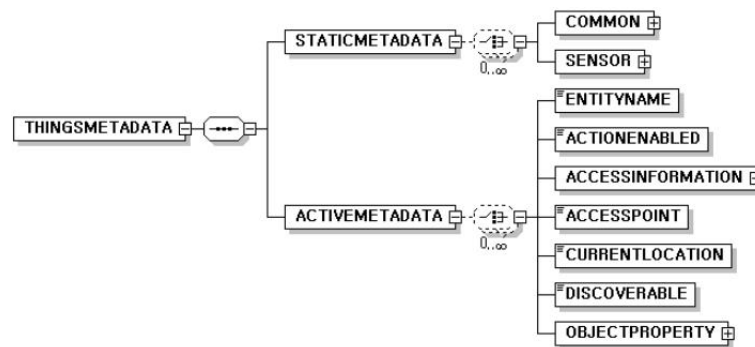
## *3.2. Elements for low-level context in sensor data abstraction*

### *3.2.1. Metadata*

Metadata is commonly used for data about data. In this paper, metadata also indicates objects in the real world, so that metadata also carries a set of descriptions; sensor device, profile, and access information. Figure 2 shows the summarized structure of the metadata and details for STATICMETADATA. Metadata contains two major fields: STATICMETADATA and ACTIVEMETADATA. STATICMETADATA field holds fixed descriptions such as device profile, vendor information, and sensing unit, etc. ACTIVEMETADATA holds changeable data which is information for accessing device, sensed value and their type information.

### *3.2.2. Sensor data acquisition model*

Collected sensor data is assembled to ACCESSINFORMATION field in ACTIVEMETADATA which contains access information to the sensor device, and sensed value type descriptions, etc. ACCESSINFORMATION field is for accessing sensor devices, which has two main types of connection: manually access and automatically access. Manually access is about to create sensor thread that handles connection procedure and its address. Automatic access happens in IoT-based sensor devices with their hub or gateway. For this IoT sensor, in general, access information is only required to the identifier and its sensor data. And the sensor data from the hub, is provided in web services or publish/subscribe based communication protocols.



**Figure 2.** Proposed metadata scheme overview.

Collected sensor data is processed with three children fields of ACCESSINFORMATION field: VALUETYPE, BASISVALUE, and GATHEREDVALUE. VALUETYPE is used for describing data types of collected data and an equation for converting sensor data, BASISVALUE is used for describing comparison information by user's choice, and GATHEREDVALUE is used for describing collected data and their processed result. For example, given a temperature sensor device from ANALOG DEVICE named TMP36, and a user wants to define the state between warm and cold by 26°C. Table 1 shows the details for which description used for among these fields.

**Table 1.** Example for describing metadata with TMP36 temperature sensor.

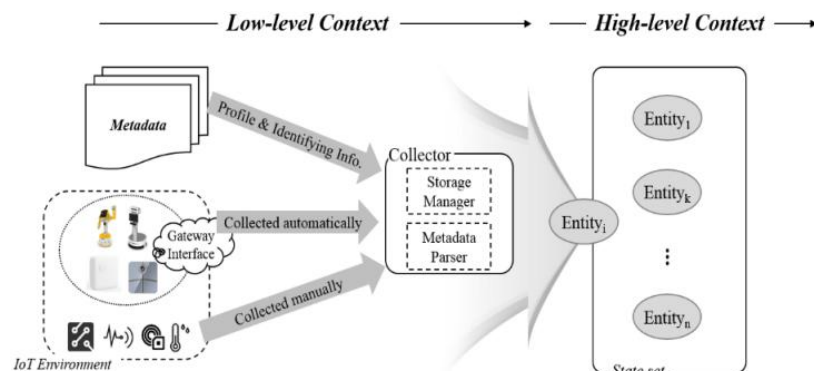
	Field	Valuetype	Basisvalue	Gatheredvalue
Generic element	Typename	Temp26	Temp26	Temp26
	Transform	true	Warm, Cold	Warm
	Equation	$((x1*5)/1024)-0.5)*100$	26.0	26.3
	Comparison	Greater	Greater	Greater
	Valuelength	1	1	1
	Valuelist	x1	var_temp	var_temp
Has type	Type	singletype	singletype	singletype
	Singletype	var_temp	var_temp	var_temp
	Complexity	N/A	N/A	N/A

For users, they tend to describe VALUETYPE and BASISVALUE for the decision making on the certain temperature degree. VALUETYPE is related to raw data description which contains the Celsius converting formula in the Equation section of VALUETYPE. And user chooses the 26°C information in the Equation section of BASISVALUE. The user wants to figure out Warm state when temperature is over 26°C. Since the Transform section of VALUETYPE is marked as true, "Warm" or "Cold" expression in the Transform section of BASISVALUE is used when the temperature information is changed. The operator "greater" is described in every Comparison section, that means processed result value will be followed whether true or false with this operator. Collected and processed sensor data will be put in GATHEREDVALUE along with the VALUETYPE and BASISVALUE contents. Through this mechanism, entity will contain the identifier, which is sensor name, and state information with collected data which is "Warm" state.

#### 4. Experiments

We describe processing elements that are explained in section 3. Procedures for generating low-level context with the proposed model is shown in figure 3.

Figure 3 shows the conceptual procedure to compose low-level context to high-level context. In figure 3, sensor devices are presented that communicate with the collector. The collector handles accessing devices, merges sensor data and their metadata, and returns entities which is the input data for high-level context.



**Figure 3.** Procedures for generating low-level context with the proposed model.

*Step 1. Initialize.* Initialize step performs creating objects for the collector's use from the described metadata input. At this point, validation check for the input data also proceeds.

*Step 2. Data acquisition & Combine.* In this step, the collector collects sensor data through the access information described in ACTIVEMETADATA field. For the IoT-based sensor devices, each connected hub will transmit the sensor data with the identifier in the form of key-value based context representation, and the collector classifies their identifier and sensor data type. Both cases will do the same combining procedure that merges sensor data and metadata described in section 3 and prepare to create the object.

*Step 3. Generate state information.* After Step 2 is done, each object contains sensor data and metadata. Metadata contains converting scheme for the sensor data with the equation form. State information is generated by the equation and the operator.

*Step 4. Return output.* When step 3 is done, each object contains sensor data and metadata with GATHEREDVALUE field filled. This output is considered as the entity and the object is transmitted to a context pool which is presented state set in figure 3.

#### 5. Conclusion

In this paper, we proposed a conceptual metadata model for the sensor data abstraction. A key idea of this model is merging sensor data and metadata in IoT environments, and the proposed model consists of elements: sensor data, metadata, and their sensor data acquisition model. With the proposed model, low-level context can be formed in structured format and it can be used for higher abstraction procedures, which are high-level context and situational relationships described in this paper.

#### Acknowledgments

This work was supported by the Industrial Convergence Core Technology Development Program (No. 10048474) funded by the Ministry of Trade, Industry & Energy(MOTIE), Korea and was supported by Basic Science Research Program (No. 2016R1C1B2009744) through the National Research Foundation of Korea(NRF) funded by the Ministry of Science and ICT(MSIT).

#### References

- [1] Atzori L, Iera A, and Morabito G 2010 *Computer Networks* **54** 15 pp 2787-2805
- [2] Li X, Eckert M, Martinez J F, and Rubio G 2015 *Sensors* **15** 8 pp 20570-607

- [3] Schilit B N and Theimer M M 1994 *IEEE Network* **8** 5 pp 22-32
- [4] Abowd G, Dey A, Brown P, Davies N, Smith M and Steggles P 1999 *Handheld and ubiquitous computing* Springer Berlin/Heidelberg pp 304-07
- [5] Perera C, Zaslavsky A, Christen P, and Georgakopoulos D 2014 *IEEE communications Surveys & Tutorials* **16** 1 pp 414-54
- [6] Maarala A I, Su X, and Riekkki J 2017 *IEEE Internet of Things Journal*, **4** 2 pp 461-73
- [7] Bettini C, Brdiczka O, Henricksen K, Indulska J, Nicklas D, Ranganathan A, and Riboni D 2010 *Pervasive and Mobile Computing* **6** 2 pp 161-80
- [8] Nambi S A U, Sarkar C, Prasad R V, and Rahim A 2014 *Internet of Things(WF-IOT) on IEEE World Forum* pp 575-80
- [9] Knappmeyer M, Kiani S L, Frà C, Moltchanov B, and Baker N 2010 *Wireless Pervasive Computing(ISWPC) on 5th IEEE International Symposium* pp 367-72
- [10] Kayes A S M, Rahayu W, Dillon T, Chang E, and Han J 2017 *OTM Confederated International Conference* Springer pp 132-53