

Context-Awareness Technology for Intelligent Sensor Networks Based on Active Mechanism

Jinhui Chen

School of Computer & Software

Nanjing University of Information Science & Technology

Nanjing, China

cjh@nuist.edu.cn

Abstract—With the emergence of intelligent sensors which have certain kinds of computing and wireless communication ability, more and more Internet of Things applications can be implemented to sense, judge and react locally according to the environment and its changes. However, mechanisms that can support context-aware mapping and that can adapt to the dynamic metadata of intelligent sensors are required. This paper analyzed context-awareness technology of the computing and applications for intelligent sensors networks, and proposed a context-awareness concept model based on active mechanism. Basing on publish/subscribe paradigm, novel realization solutions for acquiring context awareness to support this concept model are also discussed. This paper also implemented a simple demonstration to show how to use this mechanism to build system and how this system improves sensor data interpretation, reuse, and sharing in static and dynamic settings.

Keywords—Context Awareness; Intelligent Sensor Networks; Active Mechanism; Model

I. INTRODUCTION

Context awareness refers to the idea that computers can both sense, and react based on their environment. Devices may have information about the circumstances under which they are able to operate and based on rules, or an intelligent stimulus, react accordingly. Through Internet of Things, Sensors in Wireless Sensor Networks are applied to the real-time monitoring and changing the state of the physical world. However, these sensors are only used to be basic input and output devices, logic operations are performed in the background computers. At present, there are more powerful sensors that integrate microprocessors and communication modules. Therefore, they have stronger data processing and wireless communication capabilities. Such sensors are usually called intelligent sensors. In addition to having more computing power than traditional sensors, intelligent sensor also provides programming interfaces that make it possible to develop the front-end-node-oriented applications of Internet of Things. Therefore, intelligent sensors are able to sense environmental conditions and then make simple analyses and communicate, either on request or autonomously and dynamically, they play an important role in building context-aware systems.

Viswanathan envisioned that future ubiquitous healthcare systems would be characterized by context-

aware data and patient-centric decision-making, and proposed an autonomic resource framework that could harness under-utilized computing resources in the vicinity to acquire context awareness[1]. Byun presented a situation-based self-adjusting scheme, an event-based self-adjusting sensor network, and hardware and middleware implementation[2]. Kelly reported an effective implementation for Internet of Things used for monitoring regular domestic conditions. The framework is based on combination of pervasive distributed sensing units, information system for data aggregation, reasoning and context awareness[3]. Jayaraman presented a collaborative mobile sensing framework MOSDEN that could operate on smart phones capturing and sharing sensed data between multiple distributed applications and users. MOSDEN separates the application-specific processing from the sensing, storing and sharing[4]. Lee proposed a sensor-based context-aware system with a focus on three inference processes: rule, inference and pattern driven. Using pattern data derived from five families for a working week and establishing rules and inferences[5]. Silva addressed the issue of the development of middleware for intelligent systems, using techniques from information fusion and machine learning that provided context awareness[6]. Hancke presented an overview of the state of the art with regards to sensing in smart cities. Topics include sensing applications in smart cities, sensing platforms and technical challenges associated with these technologies[7]. Sivamani proposed a vertical farm ontology. With the proposed model, the information from the Internet of things is recomposed as context information and made understandable for the other systems[8]. Camarinha-Matos reported that collaborative systems would form the warp for smart networked environments wherein humans, organizations, intelligent agents, and devices collaborate[9]. Bakillah presented a context-aware ontology-based semantic mediation service for heterogeneous sensor services. The semantic mediation service integrates rule-based reasoning to support the resolution of semantic heterogeneities[10]. Solanas introduced the new concept of smart health, which is the context-aware complement of mobile health within smart cities[11]. Perera investigated the concept of sensing as a service model in technological, economical, and social perspectives and identify the major open challenges and issues[12]. Zia proposed that wireless sensor networks were still very attractive and effective for real-time spatio-

temporal data collection for water quality monitoring applications[13].

In view of the above analysis, context aware technology for intelligent sensor networks based on active mechanism is presented. The work about some key software supporting technologies is discussed. The main contents include two parts as follows. Firstly, researchers propose context awareness concept model, and define its event and rule structure. Secondly, researchers design and implement context awareness management based on publish/subscribe paradigm.

II. CONTEXT AWARENESS CONCEPT MODEL

A. Active Mechanism System(AMS)

AMS support mechanisms that enable them to respond automatically to events that are taking place either inside or outside the application system itself. Considerable effort has been directed towards improving understanding of such systems in recent years, and many different proposals have been made and applications suggested. The distinction is made between using active mechanism for monitoring, control, and reasoning.

ECA rules is a short-cut for referring to the structure of active rules in event driven architecture and active database systems. Such a rule traditionally consisted of three parts: The event part specifies the signal that triggers the invocation of the rule; The condition part is a logical test that, if satisfied or evaluates to true, causes the action to be carried out; The action part consists of updates or invocations on the local data. The context aware-oriented ECA rule is an active mechanism modeling method in the context aware technology on wearable computing architecture. It uses ECA rules to describe the context aware process model. Also it does some modifications to the context aware process meta-model to adapt for the requirements of context aware technology on wearable computing environment.

B. Context Awareness Concept Model

Intelligent sensors are deployed in a physical environment, all their behavior can be attributed to the awareness and response to the environmental state. First, Intelligent sensors should be aware of the environmental states, then, they determine how to perform their behaviors according to the rules set in advance. However, due to the limitations of their own awareness and perception scope, intelligent sensors must rely on other sensors to be fully aware of environment states and to better adapt to the changing environment. Intelligent sensors generally can't determine the event source and the event generated time, so the context awareness is thought of as a passive process. In the beginning of system design, the rule subscribers set the response to the events. The rule subscribers do not execute the corresponding actions until they receive an event published by an event publisher.

The context aware concept model is shown in Fig. 1. According to the actual implementation, each intelligent sensor takes a different role in the process of collaborative awareness. The roles associated with context awareness process include event publishers, rule publishers, rule subscribers and event forwarders. Depending on the

different roles, intelligent sensors perform different behaviors on the environmental state. The interactions between the intelligent sensors and the environment are accomplished by publish/subscribe mechanism.

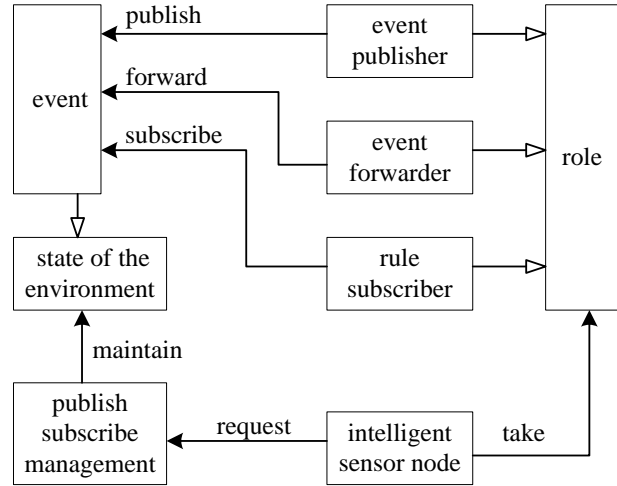


Figure 1. The context awareness concept model

C. Event and Rule in the Model

For collaborative awareness of the uncertain interactive objects, events and rules are involved. Their means are summarized as follows.

- Event contains a name and a time limitation. Name is used as a unique identifier. Time limit is used to indicate the duration of the event. When the maintain function is called, the time limit value is subtracted. When the value is less than or equal to 0, the event will be removed from the event set. When an event publisher is aware that data meet a certain condition, it will insert an event into its own event set, and then publish the event by broadcasting. When an intelligent sensor receives an event, it finds out whether there is a corresponding subscription in the local subscription set. If there is such a subscription, the corresponding action is triggered. Otherwise, the event is inserted into a local event set to avoid cyclical sending, and is published by broadcasting.
- Subscription contains a name and a callback function. The name matches an event name. The callback function indicates what action should be taken when an event is detected, and is explicitly inserted into a subscription set. The callback function is persisted locally, and doesn't have a time limitation.

III. THE PROCESS AND REALIZATION OF CONTEXT AWARENESS

A. The Process of Context Awareness

In order to realize the distributed collaborative sensing based on intelligent sensor networks, researchers can learn from the general architecture of publish/subscribe system.

In this section, researchers propose a reference process, as shown in Fig. 2, which integrates ASM with publish/subscribe architecture. According to the need of application, rule subscribers subscribe to the rules that they

are interested in. Intelligent sensor inserts a subscription into a rule set to express an interest in a certain kind of event. The event publisher monitors changes in the environment. When an event publisher discovers that the environment is in a particular state and an event triggering condition is satisfied, it will insert an event in its event set, and broadcast the event in the system. After the rule subscriber receives the event, it triggers the callback function defined in the subscription rules, and perform the appropriate actions.

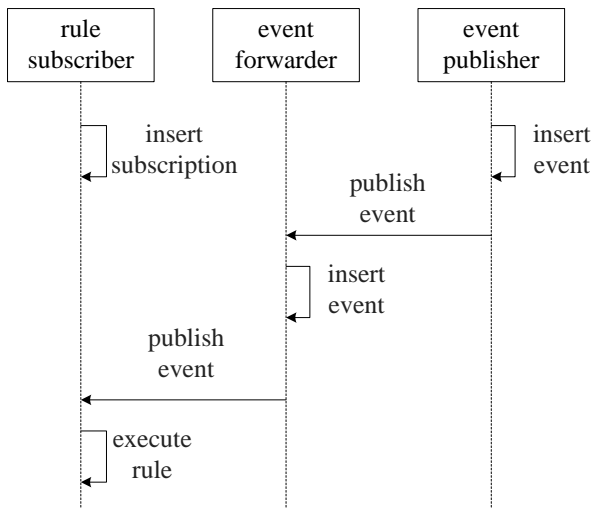


Figure 2. The process of context awareness integrated with publish/subscribe

B. The Realization of Context Awareness

In the process of collaborative context awareness, intelligent sensor involves three classes, such as event publishing, rule subscription and collection manager. Locally, each intelligent sensor manages the events and rules through its own collection manager. The class diagram of intelligent sensor networks is shown in Fig. 3.

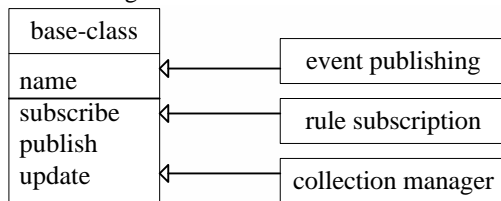


Figure 3. The class diagram of context awareness

Base-class has three basic interfaces. Other classes are the subclasses of base-class, they override three methods, such as publish, subscribe, and update methods. Therefore, they have different behaviors, and take different roles in intelligent sensor networks.

Subscribe method is used to store the rules into the local rule set.

Publish method is used to serialize the contents of event, and to propagate these events. The way of propagating events from one intelligent sensor to another may be a broadcast-based approach, or may be a unicast approach.

Update method is used to maintain the local event or rule. For publishing event class, every event has a number of variables that need to be updated over time. For

different events, Update method has different implementation strategies.

Collection manager is responsible for the management and maintenance of all kinds of events and rules. Collection manager has a unified clock. At regular intervals, it invokes the update methods of all events in turn, so as to realize the update of event state. Collection manager also provides a number of interfaces that can be invoked by developers to develop intelligent sensor networks applications. One of the most commonly used methods is the propagate-event (byte type, byte [] data), where the first parameter specifies the type of the inserted event, and the second parameter is the serialized data of the event. First, according to the first parameter, collection manager selects a specific constructor to generate different events. Then, collection manager calls publish and update methods of the new event in order that the event is saved locally and is propagated outward.

IV. TEST AND ANALYSIS

In order to show how to use this method to solve practical problems, researchers take a temperature control system as an example. The scene includes the following sensors and controllers, such as outer temperature sensor, inter temperature sensor, light sensor, window controller, air conditioning controller and central controller. The central controller has two functions: to control the window and air-conditioning switches; and to receive information sent from the sensors. The working process of the scene is as follows. The central controller receives the inter temperature information from the inter temperature sensor. When the temperature exceeds a certain threshold, the brightness information from the light sensor is received. If the brightness is relatively low, it is considered that no one is in the room, so, no adjustment is needed. Otherwise, the central controller receives the outer temperature information of the outer temperature sensor, when the outer temperature is lower than the inter temperature, it publish the event `Open_Window_Event`, on the contrary, the event `OPEN_Air_Event`. The code of the intelligent sensors and the controllers are as follows.

- For the outer temperature sensor, an event and a collection manager are defined.

```

OTS_Event= new event-publishing
(OuterTemperatureSensor);
OTS_Event.publish;
OTS_CE= new collection manager;
OTS_CE.update(OTS_Event,1);

```
- For the inter temperature sensor, an event and a collection manager are defined.

```

ITS_Event= new event-
publishing(InterTemperatureSensor);
ITS_Event.publish;
ITS_CE= new collection manager;
ITS_CE.update(ITS_Event,1);

```
- For the light sensor, an event and a collection manager are defined.

```

LS_Event= new event-publishing(LightSensor);
LS_Event.publish;
LS_CE= new collection manager;
LS_CE.update(LS_Event,1);

```
- For the controller, events, a collection manager and a subscription are defined.

```

Open_Window_Event= new event-publishing
(controller);
Open_Air_Event= new event-publishing
(controller);
Con_CE= new collection manager;
//Override subscribe method
Class Con_Rule implements rule_subscription {
    Public void subscribe() {
        int outer_temp,inner_temp,light;
        inner_temp=ITS_Event.value;
        If (inner_temp>TEMP_THRESHOLD)
        {
            light=Light_Event.value;
            If (light>LUX_THRESHOLD)
            {
                outer_temp=OTS_Event.value;
                If (outer_temp<inner_temp)
                    OPEN_Windows.publish;
                Else OPEN_Air_Event.publish;
            }
        }
    }
}

```

```

//Subscribe ASM Con_Rule
Con_Rule1 new rule_subscription(controller);
Con_Rule1.Event= OTS_Event && ITS_Event;

```

- For the window controller, a subscription is defined.

```

//Override subscribe method
Class Window_Con_Rule implements rule_
subscription {
    Public void subscribe() {
        Open the window;
    }
}

```

```

//Subscribe ASM Con_Rule
Window_Con_Rule1= new
rule_subscription(Window_Con_Rule);
Window_Con_Rule1.Event=
Open_Window_Event;

```

- For the air conditioning controller, a subscription is defined.

```

//Override subscribe method
Class Air_Con_Rule implements rule_subscription
{
    Public void subscribe() {
        Open the air conditioner;
    }
}

```

```

//Subscribe ASM Air_Con_Rule
Air_Con_Rule 1= new
rule_subscription(Air_Con_Rule);
Air_Con_Rule1.Event= Open_Air_Event;

```

As can be seen from the example, Central controller is responsible for determining when to open the window or the air conditioning, it is the event publisher. At the same time, it needs to obtain environmental information from three kinds of sensors, so, it is also a subscriber. Central controller publishes events `Open_Window_Event` and `OPEN_Air_Event`, and subscribes a rule `Con_Rule1`. Windows and air conditioning controllers are responsible for performing the actual operation after receiving the events, they are rule subscribers. Their subscription rules are `Window_Con_Rule` and `Air_Con_Rule` respectively. Three kinds of sensors are event publishers, their publishing events are `OTS_Event`, `ITS_Event`, and

`LS_Event` respectively.

V. CONCLUSIONS

In this paper, researchers propose a context awareness concept model, and event and rule are summarized in intelligent sensor networks environment. The process of context awareness are implemented based on publish/subscribe paradigm. We illustrate the active rule design through a temperature control system. The results show that the context awareness technology can easily be used, and bridge the gap between sensor nodes and applications.

ACKNOWLEDGMENT

This work was sponsored by open fund project of Jiangsu provincial research and development center of intelligent sensor network engineering technology, China (ZK13-02-03, Software technology, platform and application for sensor network).

REFERENCES

- [1] H. Viswanathan, B. Chen, and D. Pompili, "Research challenges in computation, communication, and context awareness for ubiquitous healthcare", *Communications Magazine, IEEE*, vol. 50(5), 2012, pp.92-99.
- [2] J. Byun, B. Jeon, J. Noh J, et al, "An intelligent self-adjusting sensor for smart home services based on ZigBee communications", *Consumer Electronics, IEEE Transactions*, vol. 58(3), 2012, pp. 794-802.
- [3] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes", *Sensors Journal, IEEE*, vol. 13(10), 2013, pp.3846-3853.
- [4] P. P. Jayaraman, C. Perera, D. Georgakopoulos, et al, "Efficient opportunistic sensing using mobile collaborative platform mosden", *Collaborative Computing: Networking, Applications and Worksharing, IEEE*, 2013, pp.77-86.
- [5] J. H. Lee, H. Lee, M. J. Kim, et al, "Context-aware inference in ubiquitous residential environments", *Computers in Industry*, vol. 65(1), 2014, pp.148-157.
- [6] F. Silva, C. Analide, P. Novais, "Information fusion for context awareness in intelligent environments", *Hybrid Artificial Intelligent Systems, Springer Berlin Heidelberg*, 2013, pp.252-261.
- [7] G. P. Hancke, G. P. Hancke Jr, "The role of advanced sensing in smart cities", *Sensors*, vol. 13(1), 2012, pp.393-425.
- [8] S. Sivamani, N. Bae, and Y. Cho, "A smart service model based on ubiquitous sensor networks using vertical farm ontology", *International Journal of Distributed Sensor Networks*, 2013.
- [9] L. M. Camarinha-Matos, H. Afsamanesh, "Collaborative systems for smart environments: Trends and challenges", *Collaborative Systems for Smart Networked Environments, Springer Berlin Heidelberg*, 2014, pp. 3-15.
- [10] M. Bakillah, S. H. L. Liang, A. Zipf, et al, "A dynamic and context-aware semantic mediation service for discovering and fusion of heterogeneous sensor data", *Journal of Spatial Information Science*, vol. 6, 2015, pp.155-185.
- [11] A. Solanas, C. Patsakis, M. Conti, et al, "Smart health: a context-aware health paradigm within smart cities", *Communications Magazine, IEEE*, vol. 52(8), 2014, pp.74-81.
- [12] C. Perera, A. Zaslavsky, P. Christen, et al, "Sensing as a service model for smart cities supported by internet of things", *Transactions on Emerging Telecommunications Technologies*, vol. 25(1), 2014, pp.81-93.
- [13] H. Zia, N. R. Harris, G. V. Merrett, et al, "The impact of agricultural activities on water quality: A case for collaborative catchment-scale management using integrated wireless sensor networks", *Computers and electronics in agriculture*, vol. 96, 2013, pp.126-138.