DESIGN OF A CONTEXT ANALYSIS MODEL ON USN ENVIRONMENT

Cheng Hao Jin*, Yongmi Lee*, Kwang Woo Nam**, Jun Wook Lee***, Keun Ho Ryu*

*Database/Bioinformatics Laboratory, Chungbuk National University, Korea
**Dept. Of Computer Information Engineering, Kunsan National University, Korea
***Electronics and Telecommunications Research Institute, Korea
*{kimsungho, ymlee, khryu}@dblab.chungbuk.ac.kr
**kwnam@kunsan.ac.kr
***junux@etri.re.kr

ABSTRACT Sensors used in many USN (Ubiquitous Sensor Network) domain applications generate a large amount of sensor stream data. The volume of sensor stream data is too huge to store the whole data and data speed is too fast to control each of them. In order to provide rapid and reliable context analysis service over sensor stream data, we propose a WHEN-DO context analysis model that supports the functionality of sliding window. This model is designed to be used as follows: If the sensor stream data satisfies condition in 'WHEN' clause, then it will execute actions in 'DO' clause in WHEN-DO context analysis model. The proposed WHEN-DO context analysis model can be applied to many other USN environment applications such as monitoring the status of a building and then taking actions in corresponding context condition.

KEY WORDS: Context Analysis, Context Model, Stream, WHEN-DO

1. INTRODUCTION

Sensors are used for the purpose of control, for collecting surrounding context information or monitoring the situation in dangerous places. With advances in sensor technology, current researches on the pertinent techniques are actively directed toward the way which enables the USN (Ubiquitous Sensor Network) computing service [Dey et al., 1999][Schilit et al., 1994][Weiser, 1991]. In USN environment applications, users equips with a countless number of compact wireless sensor nodes and attaches sensors to anything at anywhere and sensors collect surrounding environmental information such as temperature, humidity, light, atmospheric pressure and so on. There are a lot of USN applications like physical distribution, circulation, environment, disaster prevention, traffic, home network, automation, security and so on. By doing so, it provides us more convenient and safer life.

One of important fields in USN applications is a context service analysis that provides useful information to users according to the sensed context information. However, stream data sensed from USN is characterized as high-speed, continuous, real-time and infinite. Due to such unique characteristics of sensor data, a finite-sized buffer may not accommodate the entire incoming data, which leads to inevitable loss of data, and requirement for fast processing makes it impossible to conduct a thorough investigation of data.

Hence, in order to provide rapid and reliable context analysis services over sensor steam data, a context analysis model which supports functions of sliding window is needed. In this paper, we propose a WHEN-DO context analysis model which can be fit for USN environments. This model is designed as follows: If collected sensor stream data satisfies the condition in 'WHEN' clause in WHEN-DO context analysis model, then execute the action in 'DO' clause in WHEN-DO context analysis model.

The remainder of this paper is organized as follows. In the following section, we review related works. The proposed WHEN-DO context analysis model is presented in Section 3. The system architecture, where the WHEN-DO context analysis model being used, and detailed algorithm used in this system are shown in Section 4. We describe the system implementation result in Section 5 and the conclusion is described in Section 6.

2. RELATED WORK

U-healthcare belongs to one of USN applications and it uses medical sensors to monitor health condition of patients. Medical sensors collect physiological characteristics such as body temperature, heart rate, blood pressure and blood and so on. One of the frameworks about U-healthcare is a SAPhyRA(Stream Analysis for Physiological Risk Assessment) [Apietti et al., 2007] that performs real-time stream analysis on physiological signals to assess people health risk conditions. It learns the common and uncommon behaviours by analyzing historical data and then creates a risk model which is going to be used in real-time stream analysis. The risk level is determined by analyzing all available physiological measures in a sliding window. Risk is computed by means of formula shown in [Apietti et al., 2007]. The result of a higher risk values means a higher danger level. In SAPhyRA framework, time window size takes an important role in that if the time window is short, risk detection is instantaneous and ignores past value. However, larger size of time window can correctly detects a high risk situation when there is an anomalous behaviour for a long time in the real-time stream sensed from medical sensors. Hence, we can take rapid action to the patient according to these risk values.

One of the projects which support stream analysis is TelegraphCQ[Chandrasekaran et al., 2003]. TelegraphCQ is focused on meeting the challenges that arise in handling streams of continuous queries over high-volume, highly-variable data streams. To process the data streams, whose volume is very large and length is unbounded, operations should be run over finite windows on the streams. TelegraphCQ also supports sliding window to deal with stream data. It uses a for-loop construct to declare the sequence of windows over which the user desires the answers to the query.

3. WHEN-DO CONTEXT ANALYSIS MODEL

A well designed context analysis model affects the result of context analysis service. In this section, we describe our proposed WHEN-DO context analysis model. Figure 1 shows the structure of WHEN-DO context analysis model.

```
1. ContextModel ::= 'WHEN' <ContextCondition>  'DO' <ContextAction>
2. ContextCondition ::= <ContextExpression> [<WindowExpression>]
3. ContextExpression ::= 'CONTEXT''(' <ObjectName> ',' <ContextType> ',' <Observation> ')'
4. ContextType ::= (<AggregateOperator> '(' <TypeName> ')')  | <TypeName> | 'ACTIVITY'
5. AggregateOperator ::= 'SUM' | 'AVG' | 'MIN' | 'MAX' | 'COUNT'
6. Observation ::= (<NonStateOperator> ',' <Literal>) | (<StateOperator> ',' <State>)
7. NonStateOperator ::= <RelationalOperator> | <VariationOperator>
8. RelationalOperator ::=  '>=' | '>' | '<=' | '<' | '==' | '!='
9. StateOperator ::= 'IS' | 'ISNOT'
10.VariationOperator ::= 'RISING' | 'FALLING'
11.State ::= 'USING' | 'RUNNING'
12.WindowExpression ::= '[' 'RANGE' <IntegerLiteral> <TimeUnit> 'SLIDE' <IntegerLiteral> <TimeUnit>']'
13.TimeUnit ::= 'HOURS' | 'HOUR' | 'MINUTES' | 'MINUTE' | 'SECONDS' | 'SECOND'
14.ContextOperator ::= 'AND' | 'OR'
15.ContextAction ::= <ActuateExpression> {{<ContextOperator> <ActuateExpression>}
16.ActuateExpression ::= 'ACTUATE''(' <JavaLib> <ActuateCommand> ')'
17.ActuateCommand ::= 'ON' | 'OFF'
```

Figure 1 WHEN-DO context analysis model

At 1st line, WHEN-DO context analysis model consists of mainly two parts 'WHEN' clause ContextCondition and 'DO' clause ContextAction. At 2nd line, ContextCondition::=<ContextExpression>[<WindowExpression>] is used to describe the condition using ContextExpression and WindowExpression. ContextExpression in line 3 is about context information with ObjectName , ContextType and Observation. ObjectName is an object name. If specified ObjectName is a place, it can be one of objects with hierarchical structure. For example, assume a company there are many rooms like meeting room, seminar room and some other rooms as shown in Figure 2 and one or more sensors are located in each room.

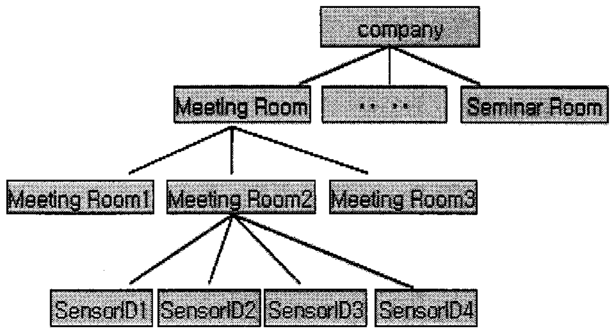

Figure 2 A company structure

If ObjectName is meeting room, here it includes meeting rooom1, meeting room2 and meeting room3. And if ObjectName is company, in this case it includes all of rooms in that company. ContextType in line 4 is the type of stream data. If sensors are capable of detecting temperature, humidity and light, then the value of contextType is temperature, humidity and light respectively. It also can be one of aggregation operator among SUM, AVG, MIN, MAX and COUNT. Observation, listed from line 6 to 11 is used as a comparison operator in math field or a state operator or a variation operator. State operator show the state of an object whether that object is being used now or not and variation operator shows the value's rising or falling. WindowExpression in line 12 has elements range and slide and they are used to deal with the stream data. The volume of stream data is very huge and speed is so fast that it should process a time window length of data at one time. After each time point a time window may slide one step, which drops off the first time point of data in the original window and appends a new window of data. Likewise, a time window can slide two steps by updating two new time points of data. Hence, range means the size of sliding window and slide means the frequencies of time window's sliding. TimeUnit in line 13 shows the time unit of sliding window. Finally, the 'DO' clause ContextAction in line 15 consists of two elements javaLib and ActuateCommand. javaLib represents the kinds of device that users are controlling and ActuateCommand represents the state of device that whether it is in on or off state. In order to understand fully at one glance, we summarize the values of each element in WHEN-DO context model in Table 1.

Table 1 WHEN-DO context model element value

ELEMENT	VALUE
ObjectName	SensorID, Room, Rooms
ContextType	Temperature, Light, Humidity, Voltage, ACTIVITY, aggregate operation
AggregateOperator	SUM, AVG, MIN, MAX, COUNT
RelationalOperator	>=, >, <=, <, ==, !=
VariationOperator	RISING, FALLING
StateOperator	IS, NOT
State	USING, RUNNING
TimeUnit	HOURS, HOUR, MINUTES, MINUTE, SECONDS, SECOND
JavaLib	Air condition, Light and other things
ActuateCommand	ON, OFF

The following is an example about WHEN-DO context analysis model.

WHEN
  CONTEXT(company, AVG(temperature), RISING, 10) (RANGE 5 minutes, SLIDE 3 minutes)
DO
  ACTUATE(Fire alarm, OK)

This means if the temperature of a room in a company is continuously going up in a high rate in a very short time, fire alarm will be executed.

4. CONTEXT ANALYSIS SYSTEM

In this section, we introduce a contextual information analysis system where our proposed WHEN-DO context analysis model is being used.

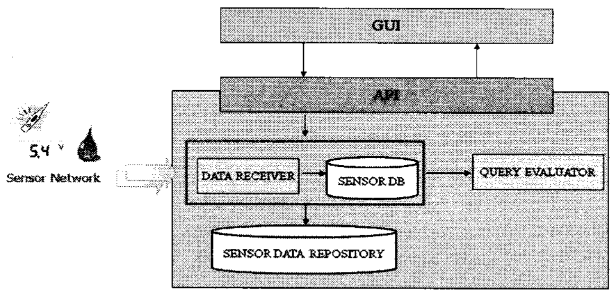

Figure 3 Context analysis system architecture

System architecture is given in Figure 3 and it consists of GUI (Graphic User Interface), API (Application Programming Interface), DATA RECEIVER, SENSOR DB, SENSOR DATA REPOSITORY and QUERY EVALUATOR.

GUI is used to create a WHEN-DO context analysis model and DATA RECEIVER receives stream data from sensor network. SENSOR DB stores sensor's information. SENSOR DATA REPOSITORY stores some stream data collected from sensors which are included in the ObjectName in WHEN-DO context analysis which is described in Section 3. QUERY EVALUATOR applies WHEN-DO context analysis model over this stream data and then determines if the collected stream data satisfies the condition in 'WHEN' clause in WHEN-DO context analysis model.

Figure 4 shows an algorithm used to evaluate query. Inputs for this algorithm are WHEN-DO context model created from GUI and information about previous sliding window. In this algorithm, we consider 4 cases because there exists 4 kinds of operators. First, we can see the value of context type temperature as AVG(temperature), because moment value is worthless to analyze. Relational operator is used to compare sensor data value in current time window with user defined value. As mentioned before in Section 3, state operate shows the object's state. The value 'ACTIVITY' and 'USING' means the object is being used now and state operator checks if sensor stream data value satisfies this predefined condition. Variation operator needs to compare the value of two sliding windows, the previous one and the current one. Calculate corresponding value and then check if sensor stream value satisfies the 'RISING' or 'FALLING' in variation operator. Finally, the actions in 'DO' clause in WHEN-DO context model are performed according to the returned Boolean value.

```
Algorithm: evaluateQuery
Input : WHEN-DO context model, previous sliding window
Output: bool
evaluateQuery(WHEN-DO, previousWindowData)
{
    double resultValue;
    if( contextType is not aggregate operation )
        see contextType as AVG(contextType);
    if(Operator is a RelationalOperator)
    {
        if( contextType is AVG(contextType))
            resultValue ← average value of current time window;
        else if(contextType is MAX(contextType)
            resultValue ← max value in current window;
        else if(contextType is MIN(contextType)
            resultValue ← min value in current window;

        booleanValue ← resultValue satisfies the condition with
        user defined value;
    }
    else if(Operator is a StateOperator)
    {
        contextType = "ACTIVITY";
        value = "USING";
        booleanValue ← check ObjectName whether
        statisfies predefined 'USING' condition;
    }
    else if(Operator is a VariationlOperator)
    {
        double resultValue1, resultValue2;
        if( contextType is AVG(contextType))
            resultValue1 ← average in previous window;
        else if(contextType is MAX(contextType)
            resultValue1 ← max value in previous window;
        else if(contextType is MIN(contextType)
            resultValue1 ← max value in previous window;
        if( contextType is AVG(contextType))
            resultValue2 ← average in current window;
        else if(contextType is MAX(contextType)
            resultValue2 ← max value in previous window;
        else if(contextType is MIN(contextType)
            resultValue2 ← min value in previous window;
        resultValue = resultValue2-resultValue1;
        booleanValue ← check whether resultValue
                        satisfy the context condition or not;
    }
    return booleanValue;
}
```

Figure 4 EvaluateQuery Algorithm

5. IMPLEMENTATION

Figure 5 shows the implementation result of our context analysis system. We implemented this system with CPU Mobile DualCore Intel Merom 1800MHz, 1G DDR2 SDRM memory and the operating system is Microsoft Windows XP professional with Service Pack 2.

We used c# language with Microsoft .NET Framework SDK v2.0 and the tool we used is Microsoft Visual Studio 2005. The dataset used to test is [Intel Lab Data, 2004]. This data collected timestamped topology information, along temperature, humidity, light and remaining voltage values once every 31 seconds.

Figure 5 consists of mainly 3 parts: WHEN-DO context analysis model, view table and the distribution of sensors.

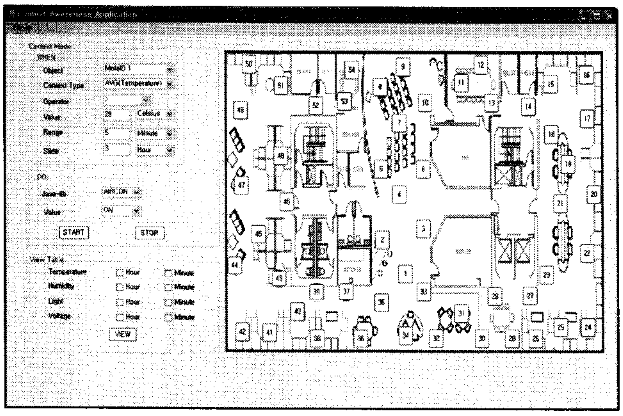

Figure 5 WHEN-DO context analysis system

The left top side of Figure 5 is WHEN-DO context analysis model where user can set all of elements in WHEN-DO context analysis model. The above settings about WHEN-DO context analysis model in Figure 5 is about that if the average of temperature collected from Mote1 is larger than 29 Celsius and continuously raises in short time unit, then the action turning on the air condition will be performed. Finally, corresponding changes will be displayed on the right side of sensor distribution part.

Bellow WHEN-DO context analysis model in Figure 5, it is a view table part that user can view data of user selected context type of average, maximum, minimum and so on. Click each sensor also can see information of that sensor's ID, place that sensor is located and the coordinates of sensor and sensing value of temperature, humidity and light.

6. CONCLUSION

USN consists of a large number of sensors and collects a large amount of data from the environment. However, these sensors generate sensor stream data that it is a large number of sequential small and tuple oriented data. It arrives in online fashion that we cannot process data in real-time. With the characteristic of data stream, the ordinary query is not fit for stream and we should use sliding window to process stream data.

In order to provide rapid context analysis service over stream data, we proposed WHEN-DO context analysis model which supports the functionality of sliding window. WHEN-DO context analysis model is mainly composed of two parts. One is WHEN part and the other is DO part. The WHEN part is used to set the condition of contextual information and the DO part is used to carry out actions when the previous WHEN conditions are satisfied. Although proposed WHEN-DO context analysis model cannot cover all of functionalities of continuous query, in some case, it is useful for applications like monitoring the status of a building and then taking actions on corresponding context condition.

REFERENCES

Apietti, D., Baralis, E., Bruno, G., Cerquitelli, T., 2007. SAPhyRA: Stream Analysis for Physiological Risk Assessment. IEEE International Symposium on Computer-Based Medical Systems, Maribor, Slovenia, pp.193-198

Chandrasekaran, S., Deshande, A., Franklin, M., Hellerstein, J., Hong, W., Krishnamurthy, S., Madden, S., Raman, V., Reiss, F., Shah, M., 2003. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In Proc. of CIDR

Dey, A., Futakawa, M., Salber, D., Abowd, G., 1999. The Conference Assistant: Combining Context-Aware with Wearable Computing. In Proc. Of the 3rd International Symposium on wearable Computers, IEEE Computer Society Press, San Francisco, pp.21-28

Intel Lab Data: http://berkeley.intel-research.net/labdata/, 2004

Schilit, B., Adams, N., Want, R., 1994. Context-Aware Computing Applications. In Proc. Of the 1st International Workshop on Mobile Computing Systems and Applications

Weiser, M., 1991. The computer for the Twenty-First Century. Scientific American, pp.94-100

ACKNOWLEDGEMENTS

This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation and also supported by IT R&D program of MKE/IITA, Rep. of Korea [2006-S-022-01, Development of RFID/USN Middleware Platform Technology].