DESIGN AND IMPLEMENTATION OF U-GIS EVENT PROCESSING SYSTEM

Moon-Soo Lee, Ju-Wan Kim

Telematics Research Laboratory
Electronics and Telecommunications Research Institute (ETRI)
{mslee, juwan} @etri.re.kr

ABSTRACT: Urban is more intelligent continuously with the help of the convergence with IT technology. And it requires an integrated control system, which can manage urban facilities or monitor large-scale events based on GIS data, to provide its citizen with various ubiquitous services such as u-Health, u-Traffic, context-awareness etc. In order to realize the intelligent city geo-sensors that have the functionalities of generic sensing as well as location awareness will be established everywhere in the near future. Our system we presented have a rule engine to handle a atomic event as well as complex events that contain control flow or branch among them. And it can allow for visualization and monitoring the results through KML (Keyhole Mark-up Language) in the Google Maps. This paper describes a u-GIS event processing system that can deal effectively with u-GIS events coming from various geo-sensor data in ubiquitous computing environments.

KEY WORDS: Geo-sensor, Event, GIS, Rule Engine

1. INTRODUCTION

In recent years, unlike traditional monolithic GIS applications recent GIS works are used in a core engine among applications and their domain is expanding to that of business and other IT services. Furthermore, with the advance in computer technology and wireless sensor network (WSN), an urban community continuously is evolved to ubiquitous society called u-City or Eco-City, and installed infrastructure to gather various kinds of sensor data occurred in the cities. To realize this ideal society, it should be able to manage its facilities and offer diverse context-awareness services like u-Health, u-Home, u-Environment, context-aware services. Therefore, an integrated operations and control system for that city requires to collect and process sensor data, and publish u-Services.

Urban integrated operations and control system is the best example of systems applied to convergence between GIS with business. It will be necessary for citizens to utilize the ubiquitous services in near future. The system can collect sensor data from RFID, fixed and mobile sensors to monitor buildings, traffic congestion, environments etc. Geo-sensor that can identify location as well as the status of its surrounding may be also used extensively.

The important issue in the wide-area wireless sensor network like u-City is how to handle large volume and heterogeneous of sensor stream data. Since large-scale sensor network has hierarchical structure composing of numerous sensor nodes, gateways and applications, each layer should be able to exchange huge amount of data and events to communicate between them. To make matter worse, there are various different types of sensor nodes to monitor for specific purposes according to sensing environments. To overcome these issues a middleware is required to process the stream data or events effectively, and must allow for rapid and effortless development of u-GIS Services.

There can be atomic and complex types to treat events in the sensor network [7]. The former can be considered as a value of status in physical environments, while the latter is a sequence of atomic or its sub-complex events. There are different methods to construct monitoring system according to what types of sensor events can be used. Database of them is more effective and efficient than middleware in case of atomic event like SQL trigger. But in the complex event, there is advantage and disadvantage if each event has branch sequence with spatial and temporal information.

Complex event processing (CEP) defines atomic events as well as their geo-spatial relationship as a rule. The rule is evaluated using the value of sensing data in real-time. And then CEP provides a proper service for users according to their circumstance. In general, a complex event model can be defined a rule that has a sequence and spatial-temporal constraints. In order to use various domains, the rule also is described as an explicit language considering flexibility and extensibility. In the active database, the Event-Condition-Action (ECA) rule is used to define complex events like SQL trigger [2, 6]. Its rule is composed of a trigger event that inquiries the modification status of columns in data table, conditions whether the states of specified sensors are satisfied with their constraints, an action that should be executed or invoked other rules if the condition is valid. Database is managing itself internally sensor data and can apply any type of operation such as base operation, data joining easily. But there is limit to handle complex events including a sequence of event rule concerned about the time constraints. In the middleware complex events is related with context-awareness technology to focus on how to support in implementing its services rapidly. Complex event model using Petri-Net is represented to tackle concurrent and non-deterministic distributed sensor network systems [4]. It also proposed authoring tool to provide a visual picture of the hierarchical event structure. To provide heterogeneous information fusion come from smart sensors, event based surveillance system supports a situation modelling method that a can be defined CEP rule for non-programmer [1]. Considering flexibility, interoperability, scalability of CEP system its engine can process the stream data in real-time and integrate with business applications in different platform like business process management (BPM) solutions [3]. This paper depicts the design and implementation of u-GIS event processing system for CEP coming from the heterogeneous geo-sensors. Our system including rule engine can monitor contexts and accidents happing in the city and provide proper service or response with the respect to their circumstances. The structure of the rest of this paper is as follows: Section 2 provides an overview of the proposed total system architecture. Section 3 describes the functionality of core modules. Finally Section 4 concludes the paper.

2. SYSTEM ARCHITECTURE

2.1 Architecture

Architecture of u-GIS event processing system consists of geo-sensor network, u-GIS event processing middleware and web client as shown in Figure 1.

In general, sensor network is installed for specific and static purposes such as structures monitoring, urban surveillance and vice versa. Geo-sensor network is added geo-sensors contained camera and GPS. It allows for utilizing temporal as well as spatial sensing data.

Unlike the generic sensors geo-sensor has the characteristic of a large-volume and heterogeneous sensor data because of being installed in the wide area and having position information such as WGS-84. u-GIS event processing middleware receives data from the network through TCP/IP, HTTP and web service. And the collected data will be converted a neutral format that can be integrated with other sensing data or used to various domain applications. Results of the middleware will be transmitted to the client applications with KML format.

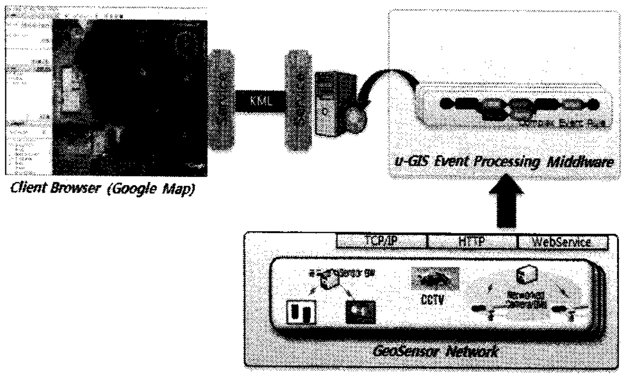

Figure 1. System Architecture

The applications download and visualize the documents having the status of the context. Google Map was used to examine the proposed system in this paper.

3. U-GIS EVENT PROCESSING MIDDLEWARE

u-GIS event processing middleware we introduce is illustrated in Figure 2. The middleware is divided to complex event processing and geo-sensor data management parts. Complex event processing is to execute deployed rules during a specific period of time. It has a memory-based database to store and read sensing states. Because of effective data management the data can be deleted automatically with accordance with their storage management policy. A key reason using off-the self databases in this paper is that complex event processing may have many rules to execute efficiently. And it also requires all of data needed in advance to evaluate them. In addition, it provides various spatial operators such as selection, join and aggregation.

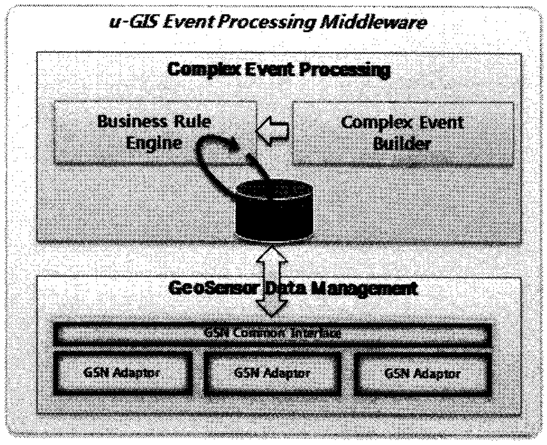

Figure 2. u-GIS Event Processing Middleware

3.1 Geo-sensor data management

Geo-sensor data management is able to register, search and manage the life-cycle of Geo-sensor network (GSN) adapters. GSN adapter can receive sensing data coming from gateways in the sensor network. The gateways may have respective interfaces and communication protocols according to what types of nodes comprised. There are two modes in receiving sensing data: push mode and request/response mode. The former sends messages in every time when an event occurs, while the latter should request and receive a sensing data of a specific node. Therefore, GSN adapter should be able to collect data regardless of the modes as a software agent. The received data can be converted to common data format and be stored into common database.

3.2 Complex event builder

There is limit to be detected contexts happening through with several sensors in the surveillance of urban space. Context–awareness in the large-scale area should be also considered spatio-temporal constraints with the values of sensors of interest. Thus, complex event can be proper method to meet the requirements. Complex event builder is to create an event rule including series of events and their spatio-temporal conditions. To allow for rapid and effortless development of u-Services, the rule enables users to integrate context-awareness or business services in other domain using web service.

The creation process as supported by complex event builder consists of three steps. The first step is to receive information about interesting sensor network from a catalog server and create grouping data table. In general, a catalog server holds metadata such as network topology, specification of nodes, update period of sensor data and vice versa. Builder reads some information of sensors location installed and that of I/O parameters to the target network and stores a database commonly used with rule engine. And then it makes spatial neighbour groups according to minimum distance among them. Usually, sensors structured in a sensor gateway are homogeneous to perform the same goal. But gateways are not. Therefore, in order to process high level of contexts or complex events abstracted data in group level based on their location should be needed.

The second step is to create a rule supporting complex events. To define an event rule user should be known about the network in detail not only what kinds of sensors are required, but also what status values of sensors are proper. A rule is decomposed into main event and sub events. Each event has input and output parameters with respected to their usage. All input variables needed to execute a rule are defined as an interface of it. They will be globally referenced by the internal events. The rule is evaluated using that variables and return results or branch the next sequence of other events. Simple events can be made by graphical user interface looks like the table of Excel editor. Each column defining a condition will be measured by "AND" logical operation. For example, a rule applied to the fire alarm will be specified that fire is occurred if temperature is above 40 degree and humidity is lower 20 percent. The rule is written by If-Then programmatic logic as shown in Figure 3.

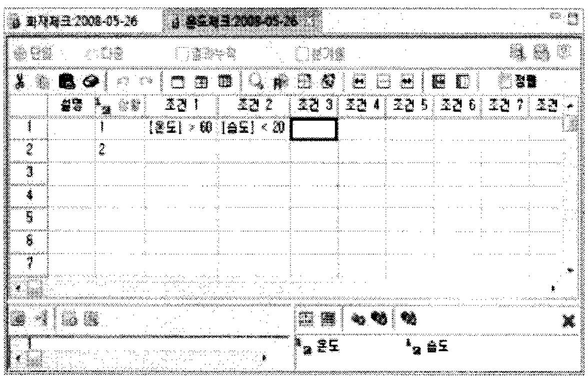

Figure 3. Example of a fire alarm rule

The final step is to register the rule created above into rule processing system. After deployment, the rule will be invoked like a web service. Thus, Complex event builder publish a rule that is defined I/O parameters and service name as a WSDL document and a KML file which can be able clients to download the evaluation results of the rule. The overall structure of the KML document is shown below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- <kml xmlns="http://www.opengis.net/kml/2.2"> -->
<kml xmlns="http://earth.google.com/kml/2.2">
 <Document>
  <Folder>
   <name>Fire Monitoring</name>
    <open>1</open>
     <Placemark>
       <name>Group03</name>
         <StyleMap>
           <Pair>
             <key>highlight</key>
            <Style>
             <IconStyle>
               <scale>1.3</scale>
               <Icon>
                <href>http://localhost:8080/KMLService/images/fired_icon.png</href>
                </Icon>
                <hotSpot x="0.5" y="0" xunits="fraction" yunits="fraction"/>
             </IconStyle>
             </Style>
           </Pair>
            ...

         </StyleMap>
    <Point>
        <coordinates>127.3707521776642,36.38236614645245,0</coordinates>
    </Point>
   </Placemark>
  </Folder>
 </Document>
</kml>
```

3.3 Rule processing system

Rule processing system is organized as rule builder service, rule DB and rule engine. The overview of the system is shown in Figure 4.

3.3.1 Rule builder service: this service is operating in the servlet container as a web application. The service can process querying, storing, testing of rules by the request of rule builder through HTTP protocols, and perform the CRUD(Create, Read, Update, Delete) job in the Rule DB.

3.3.2 Rule engine: proposed the rule engine is a run-time framework in the JAVA environments. Rule objects are executed like normal application in the Java Virtual Machine. Rule engine is composed of rule repository, repository cache, engine instance, engine factory and external interfaces. Rule repository which store and mange written rule in advance has header information to execute such as id, object name, return type, etc. Repository cache has rule data in memory to increase performance in loading rules in the rule repository every time it needed. Unlike rule information in the repository the cache has minimum data needed to be executed and not preserve management information.

Engine instance is a kind of object instance of rule with allocated resource. It is created by engine factory once in a transaction and referenced repository caches.

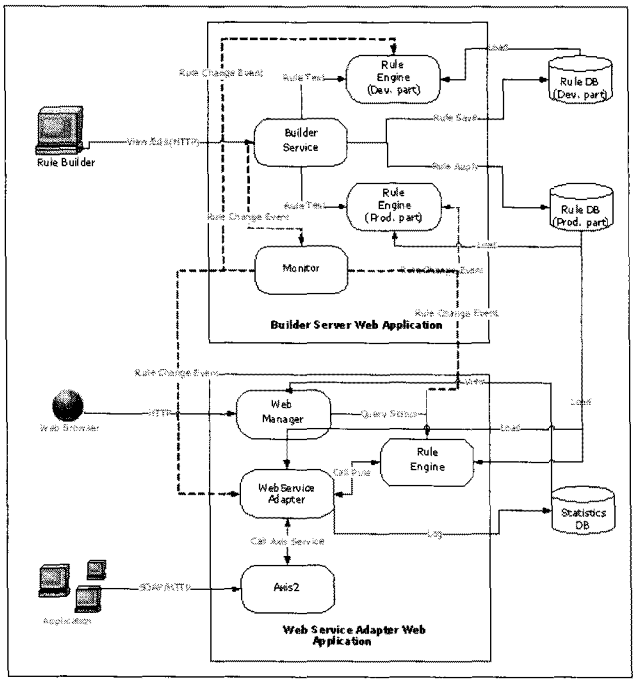

Figure 4. Rule processing system

4. CONCLUSIONS

Recently, GIS applications are positioning as a core engine to visualize other applications or to increase understanding of its functionality. They are expanding to business domain and IT services. Besides, an urban community continuously is evolved to ubiquitous society. Integration GIS with WSN is unavoidable relationship in near future. The important issue in the wide-area wireless sensor network like u-City is how to handle large volume and heterogeneous of sensor stream data. This paper describes the design and implementation of u-GIS event processing system for CEP coming from the heterogeneous geo-sensors. The system is also compatible with Google Maps client supporting KML specification.

5. REFRENCES

[1] Museux, N. Vanbockryck, J, 2007, "Event based heterogeneous sensors funsion for public place surveillance" Information Fusion, pp. 1-8

[2] U. Dayal, B. Blaustein, 1988, "The HiPAC Project: Combining Active Databases and Timing Constraints", Special Issue on Real-Time Database Systems, pp. 51-70

[3] Wei, Mingzhu, Ari, Ismail, 2007,"ReCEPtor : Sensing Complex Events in Data Streams for Service-Oriented Architectures" Digital Printing and Imaging Laboratory, pp. 1-21

[4] Binjia Jiao, Sang H. Son, 2005, "GEM: Generic Event Service Middlleware for Wireless Sensor Networks", INSS05

[5] Y. Magid, D. Oren, 2008, "Generating real-time complex event-processing applications", IBM Systems Journal, Vol 47, pp.251~263

[6] D. Zimmer and R. Unland, 1999, "On the semantics of complex events in active database management systems". In ICDE'99, pp. 392—399

[7] K. Romer, F. Mattern, 2004, "Event-based Systems for Detecting Real-World States with Sensor Networks: A Critical Analysis", In DEST Workshop on Signal. Processing in Wireless Sensor Networks at ISSNIP, Pp. 389–395

5.1 Acknowledgement

This research was supported by a grant (07KLSGC05) from Cutting-edge Urban Development - Korean Land Spatialization Research Project funded by Ministry of Land, Transport and Maritime Affairs of Korean government.