

ARCHITECTURE SUPPORTING DISCOVERY AND MANAGEMENT OF HETEROGENEOUS SENSORS FOR SMART SYSTEM USING GENERIC MIDDLEWARE

Soma Bandyopadhyay¹ and Abhijan Bhattacharyya¹

¹Innovation Lab, Tata Consultancy Services
{soma.bandyopadhyay, abhijan.bhattacharyya}@tcs.com

ABSTRACT

This Smart environments, starting from smart home to more complex one like smart city, demand efficient interoperation mechanism among different heterogeneous sensors including the discovery and the management of these devices. The diverse domains of applications also require interoperation among themselves. The middleware plays a key role to achieve this interoperation. The middleware is also responsible for providing abstractions to the application interfaces and device sensing. In the current article middleware architecture along with a method for efficient device interoperation by generating a generic device attributes (GDA) structure is presented. The middleware performs semantic analysis on the content of the device attributes while performing the discovery and managing the device. It supports, efficient way of sensor discovery, management and posting of sensed data. Smart irrigation and farming environment is considered as a use case here. The presented architecture is modular, based on object oriented concept and generic in nature. This can be further extended for any smart system. A future research scope of the proposed architecture is also discussed while concluding the article.

Keywords

Ubiquitous Computing; Heterogeneous Sensors; Middleware; Smart Environment; Adaptation; Device Discovery; Generic Framework.

1. INTRODUCTION

A smart environment essentially comprises of sensors and actuators to capture the information of the environment and to perform as per the obtained command respectively. It also encompasses the services and applications interacting with the appropriate sensors and actuators to get the information on the environment, and to act as per the situation intelligently. Therefore sensing and processing the raw sensed data and tunneling that data to the appropriate services and applications in appropriate formats are essential for building smart system. The sensors and the applications/services are heterogeneous in nature and reside in a distributed environment. A faster interoperation mechanism hiding the complexities is also an important need for any smart system.

A middleware is required, in order to achieve this requirement. It primarily acts as a bond across the heterogeneous sensors and heterogeneous applications /services. Its prime objective is to provide a homogeneous interface hiding diverse properties of heterogeneous sensors as well as applications/services.

This article focuses on design and development of a generic object oriented middleware architecture, proposes a novel method for efficient device management, as well as faster way to interoperate with the sensing devices which are heterogeneous in nature, using a method of generating generic device attributes (GDA) structure and performing semantic analysis on the

content of the device attributes while performing the discovery and managing the devices by the device manager. The top level device attributes can be expanded into layers of sub attributes maintaining a tree-like data structure. The semantic analysis on the values of the device attributes generates cluster of sensing devices and assigns them various classes. Proposed middleware supports the following properties, 1) device discovery of heterogeneous devices and forming a novel data structure with generic device attribute (GDA), 2) device management of heterogeneous sensors, 3) resolution of semantics and syntaxes while interoperating among heterogeneous sensors, 4) extraction of context from the sensed data, 5) handling critical data based on the real-time requirements of the event to be processed and the state of the sensors, 6) creation of configuration schemas based on XML, 7) creation of device functionality schemas based on XML, and importantly 8) providing an application adaptation interface. The main functional components are based on the functional components as proposed in [1] for IoT (Internet of Things) middleware.

The proposed architecture, as presented here, has a layered architecture with two abstraction layers. The architecture is modular based on an object oriented model supporting various generic modules like devices, certified- sensors, sensor_adaptor etc. with clearly defined inter-module relations. The adaptation layers are flexible enough to adapt new sensors or applications/services without requiring any modifications to the core components of the middleware. The new objects inherited from the existing ones are needed to be added to the framework and corresponding modifications are to be made in the XML schemas. Thus the proposed architecture can be used as a framework for developing a middleware capable to be extended to support any smart system as it addresses the generic requirement of these systems and does not require modifications inside the core components to support diverse configurations. Only the XML schemas specific to device configuration and functionality, its class path for dynamical execution are to be modified for this purpose.

The case study, as presented here, consists of a smart irrigation and agriculture system. Device interoperation is shown based on Zigbee and Ethernet based sensing devices. The class diagram depicts the relationship among the various objects related to this use case.

The remainder of this article is organized as below. First, the related work in middleware architecture and its review is presented followed by an overview of the proposed system. The architecture of the system along with the class diagram is then described. The next section describes a practical application of the middleware in a smart irrigation and farming system with an illustration of forming GDA structure. The final section concludes this article with the future research scope on the proposed architecture.

2. RELATED WORK

The middleware functionalities are widely studied to address the important aspects and to address various challenges of setting up smart environment in the domain of ubiquitous computing as well as IoT. The role of middleware in IoT is studied extensively and the various functional components of the middleware system are proposed in [2]. This survey paper concludes the open issues and challenges about the middleware.

In [3], an interesting research project called ‘Munich’ (Mobile Users in a Non-intrusive Computing Hierarchy) on subjective sensing to meet the user needs intelligently and to support personalized services based on mobile phones with a layered architecture is presented.

In [4], an energy efficient mobile sensing system having a hierarchical sensing management schemes for setting up environment is presented. Here XML is used to define user state and user

state transition rules, and these XML based state descriptors are used as inputs to turn sensors on and off.

In [5], a declarative model for Sensor Interface Descriptors (SID) based on OGC's (Open Geospatial Consortium)[9] SensorML[13] standard is presented to close the gap in standardization for the connection between the SWE (Sensor Web Enablement) and the underlying sensor layer with heterogeneous protocols. Here a generic SID interpreter capable of connecting sensors to Sensor Observation Services and Sensor Planning Services based on their SID has been developed. The system has two components, the SID interpreter and the SID interface. The SID interpreter runs on the data acquisition system and uses SID instances for the different sensors of the sensor network to translate between the sensor specific protocol and the SWE protocols. The interpreter is responsible to register a sensor at a SWE service and to upload sensor data to an SOS. Also, it is responsible for the opposite communication direction and forwards tasks received by an SPS to a sensor.

In [6], a middleware system 'LinkSmart' is proposed. This middleware combines semantic web services technology with SOA-based principles. It provides a mechanism for wrapping standard API interfaces of sensors and various physical devices with a defined web service extension, which is enhanced by a semantic description of provided or generated WSDL (Web Services Description Language) [12] files thereby connecting the devices and their local networks to the outside world through broadband and/or wireless networks.

Ref. [7] highlights the major issues in designing the middleware for WSN like heterogeneity, complex event processing, QoS support and access prioritization, etc. It divides the state-of-the-art middlewares into some distinguished categories like Interoperability Middleware, Web Service Enabler, Semantic Sensor Web Middleware and Information Processing Middleware based on the application usage. It proposes a middleware design to address the said challenges it has identified. It proposes some generic layers as follow: connectivity layer, knowledge layer to form a knowledge-base for device management, semantic metadata, etc., information processing layer to handle the incoming queries and service provisioning layer to addresses the gap between traditional high-level application protocols and the underlying layers.

However, these research approaches do not make use of any generic middleware architecture which can be extendable to both fixed and mobile platforms to build any smart system. Importantly they do not specify about both the sensors as well as application abstractions and an object oriented based modular architecture which can be used as framework for building middleware for any smart system starting from interoperating among diverse sensors to post sensed data to application/services as per the user needs.

Another major aspect of the present article is a proposed scheme of generic device attribute structure based efficient device management using device attribute semantics while performing device discovery. Ref. [20] specifies about generic middleware architecture but does not specify about this scheme.

There are existing works to associate device descriptors with the discovered devices using different discovery protocol like in patent US7673077B2 [15] which provides a method for target discovery in an iSCSI storage area network in which the host initiator uses a target discovery manager which communicates with the target devices through a network. The target devices are discovered by the discovery manager by way of different discovery protocols in such a way that a single device can be discovered by multiple protocols. The target discovery manager functions to coalesce target devices discovered through multiple discovery protocols into a register of discovered target devices and sets priorities among the protocols to use when one or more of the listed target devices is requested by the host initiator. The target discovery

manager provides the user with a multi-discovery protocol means by which to initiate a target discovery when a desired target's location is not known or maintained in the register of enumerated target devices. So, this invention provides the host initiator with the means to identify a desired target to the target discovery manager and allow the target discovery manager to simultaneously initialize several discovery protocols at once so as to discover the requested target device as quickly and as efficiently as possible. Thus it provides a clear way to map host side enumerations of target devices to different iSCSI discovery protocols.

There are also existing works to associate unified device descriptors (UDD), like patent US6581094B1 [16] describing a method executed by one or more digital devices operating in a networked environment including process of storing network address(IP or URL) for each digital device. Unified device descriptors are created using XML for each digital device in the network. Specified attributes in a search request are matched with attributes in UDD file to render a selected digital device. Web browser is used to identify devices within a networked environment which enables linking a digital device to a network regardless of the connectivity scheme and operating system used within the network.

In the patent [17] a method, system, and service of analyzing, and discovering of electronic documents in an intranet has been provided using semantic analysis. Intranet includes multiple web sites. The method has following steps: (1) crawling HTML content and text content in a set of the sites, (2) deep-scanning non-HTML content and non-text content in the set of sites, (3) reverse-scanning the set of sites, (4) performing a semantic analysis of the crawled content and the deep-scanned content, (5) correlating the results of the semantic analysis with the results of the reverse-scanning, and (6) comparing user navigation patterns and content from the members of the set of sites. Also this patent further includes combining the results of the performing, the results of the correlating, and the results of the comparing. This patent finds application in large organization handling lot of server content data. Maintenance and management of these servers is very much expensive, thus reducing the number of servers in the system helps in effective savings for the company. It uses semantic analysis of the crawled and deep-scanned content from electronic documents to reduce data content for this purpose.

The work presented in patent US7809711B2 [17] emphasizes on the electronic files in the intranet. It deals with the data content of the files. The type of files can be html or non-html based. Thus the system does not state about smart devices. Here the electronic files are considered as devices and also does have to address much of heterogeneity.

In patent US20110022733A1 [18] provides a method and system for customized data delivery and network configuration by performing the aggregation of device attributes using a specific network architecture having multiple access gateways. In this patent each of the plurality of network devices has the following attributes - serial number or unique identifier, a manufacturer identifier, a model identifier, a hardware configuration, a software configuration, an operating system identifier, available and/or total memory, available and/or total processing cycles, security information, battery level, and/or power settings .

The patent EP2278774A2 [19] provides a method and system as mentioned in [18]. Along with that it further states about securing the aggregated device attributes utilizing IPSec and/or MACsec protocols, customization of the content is done by one or more of compressing, decompressing, down-sampling, and/or up-sampling, one or more link can be used optical link. Customized data can be exchanged once device information or capabilities are exchanged.

However, none of the prior arts talks about forming a generic structure of device attributes for the heterogeneous smart sensing devices in a smart environment during the device discovery phase, specifically, while performing device capability negotiation in an edge gateway. Also,

none of them come up with a proposal to form multiple layers of clustering of sensing devices based on semantic analysis of the contents of the device attributes. Thus none of the schemes is able to counter the problem of minimizing the time required for device prioritization and device interoperability for a required type of interaction.

3. SYSTEM OVERVIEW

The generic middleware for smart system proposed here consists of a key control layer shown as SMART Controller and two adaptation layers, sensor abstraction and application abstraction. The layered architecture is depicted in Figure 1.

Fundamentally, the middleware is responsible to facilitate the exchange of information between the remote applications and the sensors embedded in the smart environment. The remote application connects to the middleware over the Internet. The middleware receives the information about the smart environment from the sensing devices and in turn provides this information to the remote applications based on the application requirements.

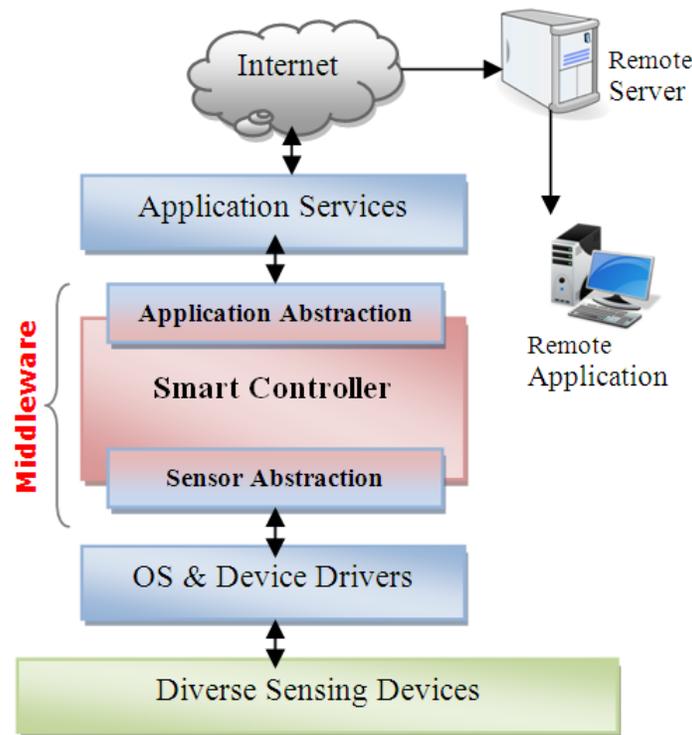


Figure 1. System Overview

The middleware sits on top of the 'OS (Operating System) & device drivers' layer and below the 'application services' layer. The 'device drivers' layer takes care of the underlying protocols for communication between the middleware system and the sensing devices. The 'application services' layer runs the local services for the remote applications.

The heart of the middleware system, the 'smart controller', which handles all the core functionalities to communicate between the sensing devices and the remote applications as well as performs different device management activities, is responsible for all the operations like

discovering the sensing devices, making the raw data presentable to the web interface, performing device monitoring, posting the data to the application abstraction layer etc.

The smart controller interacts with two abstraction layers of the middleware. At the bottom is the ‘sensor abstraction layer’ and at the top is the ‘application abstraction layer’. These interfaces play a vital role to equip the system with generic interfaces.

The bottom sensor abstraction layer provides a general abstraction with respect to the diverse sensors so that the heterogeneous sensors can be interacted with a common format. Similarly the top level application abstraction layer provides a common interface to interact with the heterogeneous applications.

Thus these two abstraction layers make the middleware adaptive to any smart environment.

4. SYSTEM ARCHITECTURE

The proposed architecture of the middleware for smart system is represented in the present section. A block diagram of the system architecture is depicted in Figure 2.

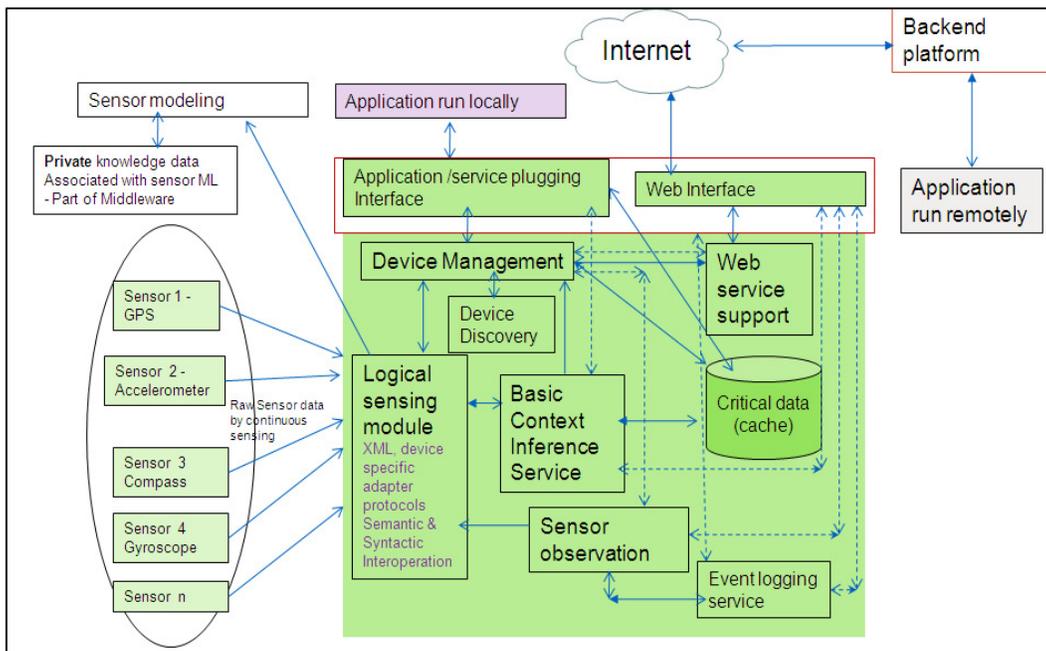


Figure 2. System Architecture

The descriptions of each module are as follow:

4.1. Logical sensing module

The core part of sensor abstraction layer, the logical sensing module collects data from various diverse sensors like location, acceleration, angular velocity, temperature, humidity, light, etc. using diverse communications and messaging protocols following different standards. This block is responsible to handle the interoperation issues, resolving the syntax and semantics of the messaging protocols by using the specific sensor adaptation protocols. It uses specific sensor descriptions standard like SensorML. This sub module fundamentally encapsulates all the details of the sensor interoperation and forms sensor abstraction layer.

4.2. Device management module

The device management module resides in the smart controller layer. It is responsible for managing all active sensing devices by activating its different sub – modules and using associated handshaking messages with them. It controls posting of the sensor data to the application abstraction layer. It consists of sub components like device discovery which scans the environment through the available interfaces and communication protocols in regular intervals to detect the new sensing devices. It does a capability negotiation based on the various attributes like interface, protocol etc. of the sensor devices. Device discovery sub module executes SCAN command periodically. It broadcasts a handshake message to all the sensors, and accepts responses from the active sensing devices. It uses the device adapters and the associated protocols for those sensing devices made available during system configuration. Device management module maintains a device management table for the sensing devices having device-type; manufacturer identifier, Interface type (Zigbee, Bluetooth), protocol identifier and the associated service identifier. Some of the device specific information is taken from the configuration file like device type, interface type, and some of the information are generated dynamically like unique device/sensor identifier during device discovery and are maintained in the device table. It collects the inputs related to basic device properties from the XML based configuration schemas. A state flag is maintained by the device manager in the device management table depending on the response received from the sensing device during the periodic scanning. Figure 3 illustrates the state diagram for the sensors.

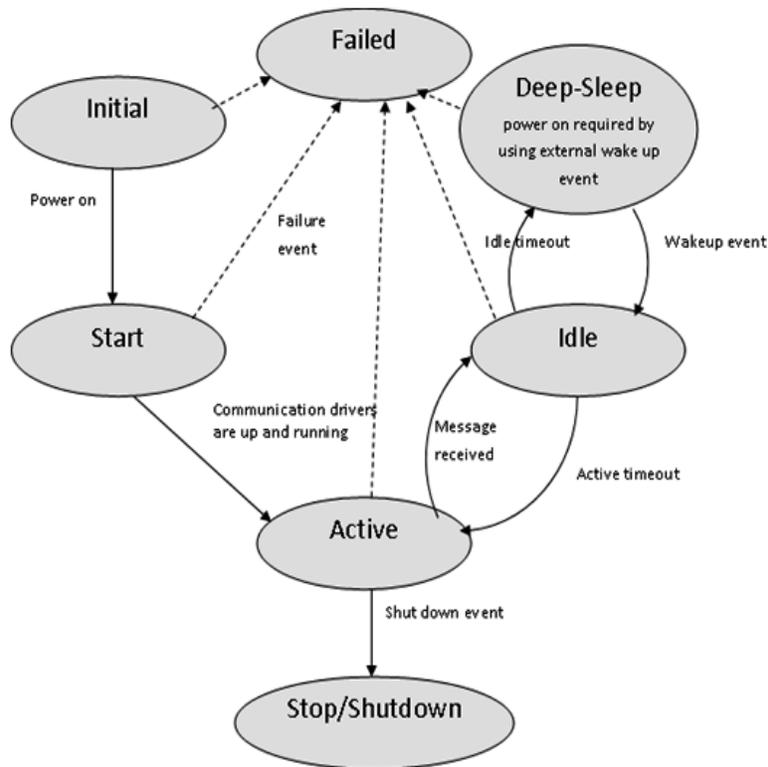


Figure 3. State Diagram of Sensors

Respective events are triggered as per the application needs and a reduced set of sensors is generated in each scan depending on the sensor state idle or deep-sleep. The device management table also maintains a service identifier associated with the device. The other two important sub blocks of the smart controller layer are sensor-observation and context inference service which closely interact with device management module. The major activity of these two blocks is to collect data from sensors.

4.2.1. Formation of the GDA

A major portion of this work has been dedicated to come up with a generic approach for enhancing the device interaction with the help of forming a data structure based on generic device attributes (GDA). The GDA structure is created during the device capability negotiation phase. The algorithm performs semantic analysis on the content of the device attributes while performing the discovery. The top level device attributes can be expanded into layers of sub attributes maintaining a tree-like data structure. The semantic analysis on the values of the device attributes generates cluster of sensing devices and assigns them various classes. This analysis provides a faster way to interoperate with the sensing devices which are heterogeneous in nature.

The device manager of the middleware creates a GDA for the sensing devices during discovering those devices, specifically while performing the device capability negotiation, in the process of device discovery. GDA is maintained by the device manager throughout the active lifespan of the devices. The GDA provides a hierarchical structure of attributes for the smart sensing devices. Attributes are further divided into the sub attributes, for example protocol is one of the attributes which can be divided into two sub attributes

1. Communication [may have the value of WiFi, Zigbee, Ethernet, Bluetooth etc.]
2. Syntax and semantic resolution.

The attribute 'communication' again can be further divided into

- device-identifier,
- transmission-rate,
- duplex-mode.

'Syntax and semantic resolution' can be further divided into encoding and decoding scheme.

A GDA based tree is presented in Figure 4.

The attributes and sub attributes are presented with different levels in this figure.

Level 1 proposes the primary classes of attributes, Level 2; Level 3 describes the further division of the attributes from Level1 and Level2. The attributes in level 3 corresponding to their parent are presented as exemplary set and are not limited to the list of sub attributes presented here; the sub-attributes may be extended further as per the need of the application. For example the sub attributes of 'Sensing Category' belonging Level 3 may be extended further. The device manager performs semantic analysis on the content of the GDA of the sensing devices. This results the formation of device clusters categorized by its first level of hierarchy. The classes of devices can be further generated by combining the various device clusters depending on the sensing category of the GDA.

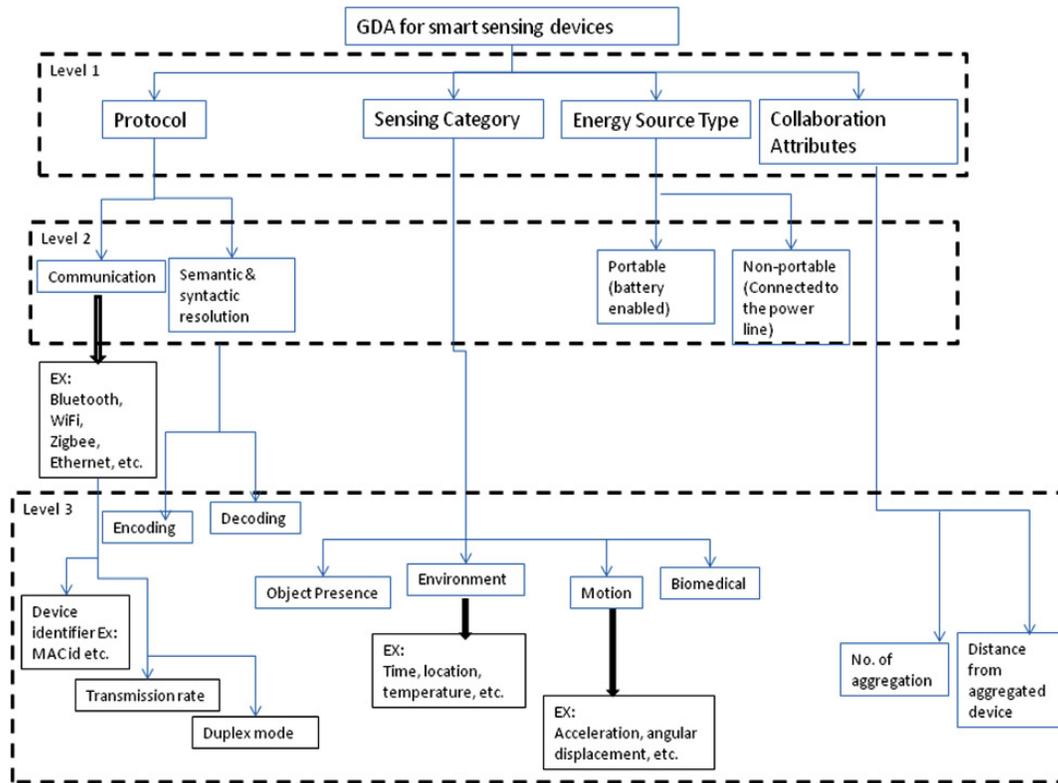


Figure 4. The GDA structure

Figure 5 provides a flow chart of sample execution considering only 2 levels of Semantic Analysis (SA) to create hierarchical clusters to perform optimized device interactions.

4.3. Sensor Observation Module

Sensor observation module interfaces with the logical sensing module to extract and post the data to the application. It gets an event from the device manager to start the posting of data. The context inference service block is part of smart controller layer. It retrieves contextual information both in synchronous and asynchronous mode after getting a triggering event from the device management module, which defines a structure for contextual data using the device properties defined in XML, and the demand of application/services. A tiny storage space is used as critical data cache to keep the critical data, mainly the contextual data, depending on the real-time requirement of the processing of that data as well as event identifier with applications/services – sensing-device mapping. Event logging module is interfaced with the sensor observation module. It keeps on posting of sensor observations as per the application’s demand through web interface. It uses publish-subscribe model.

4.4. Application Abstraction Module The applications abstraction layer consists of the web interface and application plug-in interface. The application plug-in interface provides mechanisms to register applications running locally on top of the middleware layer. Web interface supports the interfacing with the remote applications. Both these sub modules interact with the device management layer to post their demand as well as get the notifications related to the sensing devices from the smart controller layer via the device management module. Different OGC (Open Geo spatial Consortium) [9] based services can be used as a remote service and its counterparts as local applications.

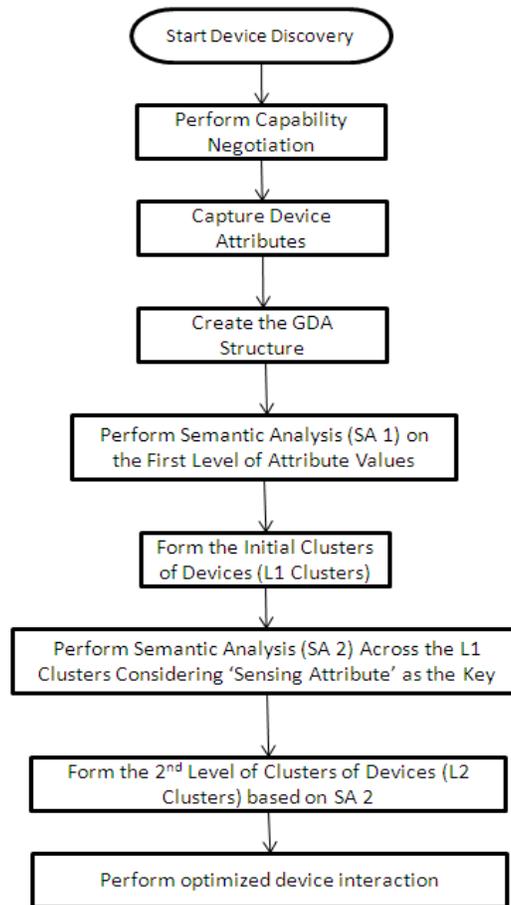


Figure 5. Execution flow

Figure 6 depicts the class diagram of the object oriented model used for designing the architecture. The various modules and their relationship are depicted in this figure. As an example it can be observed, device-operation, device-manager, devices, sensors, actuators, adapter, sensor-adapter are various modules/classes or interfaces where sensors and actuators are inherited from the device class, sensor-adapter is inherited from adapter class etc.

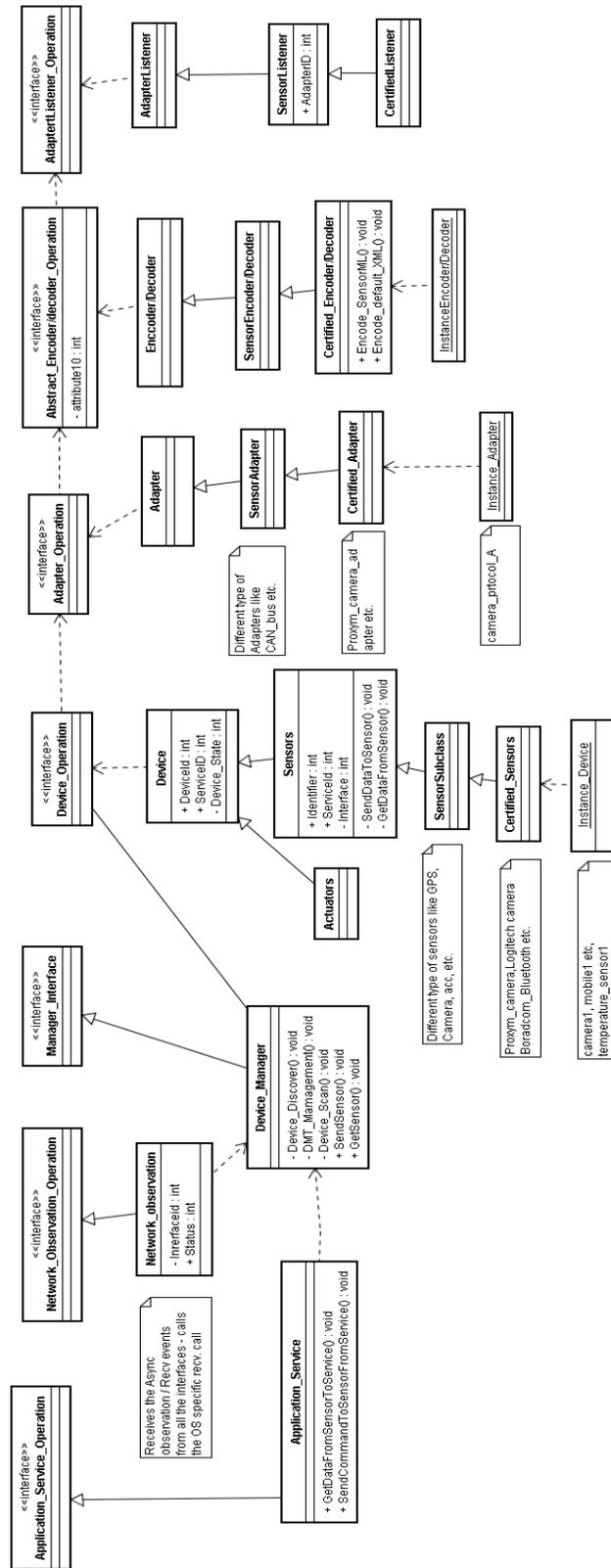


Figure 6. Class Diagram

5. USE CASE: SMART IRRIGATION AND FARMING

This section describes the application of the proposed middleware in the context of a smart irrigation and farming system as a smart environment [10]. This section also describes how the GDA can be formed out of the associated sensors.

The smart irrigation and farming system ‘senses’ the different parameters from the environment using a composition of remote and in-situ sensors. The different heterogeneous sensors are used to measure temperature, rainfall, humidity, solar radiation, soil moisture, location, time, and also to detect presence of objects using remote camera.

Figure 7 depicts the functional units of the use case showing the farmland equipped with the above mentioned heterogeneous sensors. The remote application service station is the unit where the relevant applications reside. This service station can be a mobile phone or a PDA (personal digital assistant) running suitable applications or it may be remote monitoring office monitoring the state of the environment and triggering necessary measures. The applications interacting with the middleware extracts the sensed data by using the sensor abstraction of the middleware and provide appropriate information to the farmers. An example of such information can be a message suggesting the farmer about the best suitable crop to choose for the prevalent environment. Another example application can be triggering security alarms in case unwanted cattles have intruded into the land. Another such useful application may be triggering the necessity to irrigate the land by measuring an inadequate soil moisture level and so on and so forth.

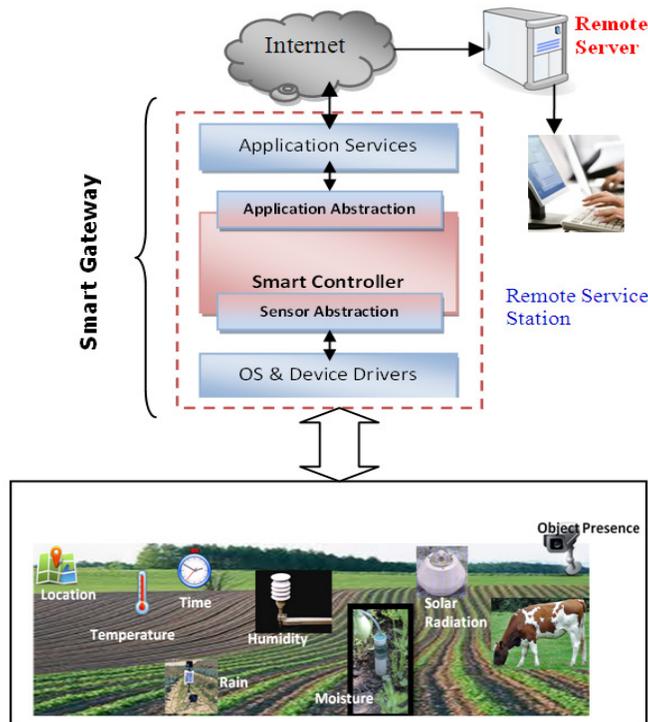


Figure 7. Smart Irrigation and Farming Environment

Again, as it is evident that the list of sensors is not exhaustive, the system may need to cope up with new types of sensors. This enhancement can be easily done through the sensor abstraction

layer by adding new sensors. Similarly new monitoring applications can be interfaced with middleware using its application abstraction.

Figure 8 shows an excerpt from a typical device configuration schema written in XML. This configuration takes care of a Zigbee based temperature sensor and an Ethernet based remote camera for detecting presence of objects (e.g., cattle) in the farmland respectively. A close observation of Figure 8 shows that the sensor class provides encapsulation for the device specific 'Xbee' [8] module.

```

<?xml version="1.0" encoding="UTF-8" ?>
<Class>
<Device>
<Type>Zigbee</Type>
<Name>com.rapplogic.xbee.api.XBee</Name>
<Protocol>Xbee</Protocol>
<AdapterC>org.middleware.Xbee_Sensor_Adapter</AdapterC>
<DXMLPath>C:\\eclipse-java-galileo-SR2-win32\\New
Folder\\edge\\zigbee_descriptar.XML</DXMLPath>
</Device>
<Device>
<Type>Ethernet</Type>
<Name>org.middleware.Devices</Name>
<Protocol>RFID</Protocol>
<PacketSize>1000</PacketSize>
<AdapterC>org.middleware.Xethernet_Sensor_Adapter</AdapterC>
<Hostname>"01hw185578.India.TCS.com"</Hostname>
<DXMLPath>C:\\eclipse-java-galileo-SR2-win32\\New
Folder\\edge\\Ethernet_descriptar.XML</DXMLPath>
</Device>
</Class>
    
```

Figure 8. Excerpt from the XML based sensor abstraction

5.1 Formation of the GDA

In this section an example of device clustering is produced. Let the following table provides an example set of sensing devices.

Table 1. List of Sensors.

Sensor Type	Identifier	Quantity
Temperature Sensor	T	1
Rainfall Sensor	R	1
Humidity Sensor	H	1
Solar Radiation Sensor	SR	1
Soil Moisture Sensor	SM	2
Location Sensor	L	1
Time Sensor	t	1
Camera	C	1
Presence Detection Sensor	P	1

Different sensors will be equipped with different communication techniques and different syntactic and semantic schemes. Even a similar sensor from two different vendors can have different semantics.

Now, referring to the tree of attributes in Figure 5, Table 2 can be formed.

Table 2. Table representation of the GDA tree

Device ID	Protocol		Sensing Category	Energy Source		
	Comm ⁿ	Sym & Syn		Portable	Non-portable	# of aggregation
T	C1	S _{T1}	Environment: Value = Temperature	Yes	No	0
R	C1	S _{R1}	Environment: Rain	Yes	No	0
H	C2	S _{H1}	Environment: Humidity	Yes		0
SR	C3	S _{SR1}	Environment: Solar Radiation	Yes	No	0
SM#1	C2	S _{SM1}	Environment: Soil Moisture	No	Yes	0
SM#2	C1	S _{SM2}	Environment: Soil Moisture	Yes	No	0
L	C2	S _L	Environment: Location	Yes	No	0
C#1	C1	S _{C1}	Object Presence	Yes	No	0
C#2	C2	S _{C2}	Object Presence	Yes	No	0
P	C4	S _{C2}	Object Presence	Yes	No	0

Here, C<n> is some typical communication protocol and S<str> is a typical syntactic & semantic rule.

5.2 Example formation of the clusters

Let A, B, C, D, E present the different semantic rules to generate the respective device clusters. Considering a temperature sensor as in the first row of the above table the cluster entries for each rule can be created as shown in Table 3.

Table 3. Sample device clusters

Rule Name	Rule	Cluster Entry
A	SensingCategory.Level3attributes.value + Protocol.communation.value	(T, C1)
B	A+ Protocol.Semantic&syntactic resolution.value	(T,C1,ST1)
C	B +Energy Source Type.value	(T, C1, ST1, 0)
D	B +Collaboration/aggregation Attributes .value	(T, C1, ST1, 0)
E	C+ Energy Source Type.value + Collaboration Attributes .value	(T, C1, ST1, 0, 0)

5. CONCLUSION

In this article generic middleware architecture to interoperate with diverse sensing devices and diverse domain of applications in a smart system is presented. It shows interactions among every major block of the system. Heterogeneous sensors management based on the various states of the sensors is one of the challenges addressed here. The proposed architecture uses various configuration schemas and adding a new sensor requires the addition of its properties in the appropriate XML schemas therefore it becomes extensible for different environments. The proposed architecture follows 'object-oriented' concept and the presented UML based object oriented model gives a clear view of different class components of the system. It proposes an algorithm for creating a hierarchical device attributes structure, and to form a device clusters to achieve faster device interoperation. The modular and object oriented architecture eases its development using Java and running it as a service of JVM (Java Virtual Machine) as well as OSGi(Open System Gateway Interface)[14] based platform which will be easy to port and implement.

The use case scenario on smart irrigation and agriculture system describes how the generic and adaptable characteristics of the middleware help to address the interoperation; mainly the semantic and syntactic, and device management issues.

There are multiple scopes for future research works like, 1) to explore how to meet the real time requirements of interoperation considering the facts of energy constraints of the sensing devices, 2) optimized communication mechanism with ensured reliability particularly when the middleware runs on a resource constrained device , 3) optimized event handling mechanism to be adopted by the middleware's smart controller to actuate sensing/actuating device 4) to define the methods to achieve faster service collaboration providing sensing-device and service/application mapping. Currently we are conducting research on the 2nd and 3rd points mentioned above.

REFERENCES

- [1] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti, Subhajit Dutta “A Survey of Middleware for Internet of Things” CoNeCo 2011, Ankara, Turkey, June 26 - 28, 2011.
- [2] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti and Subhajit Dutta “ROLE OF MIDDLEWARE FOR INTERNET OF THINGS: A STUDY”, International Journal of Computer Science & Engineering Survey (IJCSSES) Vol.2, No.3, August 2011.
<http://airccse.org/journal/ijcses/papers/0811cses07.pdf>
- [3] <http://research.microsoft.com/pubs/135844/SubjectiveSensing-statement.pdf>.
- [4] www.usc.edu/dept/ee/scip/assets/002/63910.pdf.
- [5] Arne Broering, Stefan Below and Theodor Foerster, “DECLARATIVE SENSOR INTERFACE DESCRIPTORS FOR THE SENSOR WEB”, Proceedings of WebMGS 2010: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services, Como, Italy, August, 2010.
- [6] Peter Kostelník, Martin Sarnovský, Karol Furdík “The Semantic Middleware for Networked Embedded Systems Applied in the Internet of Things and Services Domain”, Scalable Computing: Practice and Experience (SCPE). Scientific International Journal for Parallel and Distributed Computing. Volume 12, Number 3, September 2011, pp. 307–315.
- [7] Frieder Ganz, “Designing Smart Middleware for Wireless Sensor Networks”, The 12th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, PGNET2011, The School of Computing and Mathematical Sciences, Liverpool John Moores University, April, 2011.
- [8] <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/xbee-series1-module#overview>
- [9] <http://www.opengeospatial.org/standards/>
- [10] Evaluation of Soil Moisture-Based on-demand Irrigation Controllers: Final Report”, Prepared for and funded by: Southwest Florida Water Management District, Aug, 2008.
- [11] http://www.knopflerfish.org/snapshots_trunk/current_trunk/docs/android_dalvik_tutorial.html
- [12] <http://www.w3.org/TR/wsd1>
- [13] [.http://www.opengeospatial.org/standards/sensorml](http://www.opengeospatial.org/standards/sensorml)
- [14] <http://www.osgi.org/Main/HomePage>
- [15] US7673077B2 2010-03-02 Multi-protocol iSCSI device discovery for on demand device enumeration.
- [16] US6581094B1 2003-06-17 Apparatus and method for identifying a digital device based on the device's uniform device descriptor file that specifies the attributes of the device in a XML document in a networked environment.
- [17] US7809711B2 2010-10-05 System and method for semantic analysis of intelligent device discovery.
- [18] US20110022733A1 2011-01-27 CUSTOMIZED DATA DELIVERY AND NETWORK CONFIGURATION VIA AGGREGATION OF DEVICE ATTRIBUTES.
- [19] EP2278774A2 2011-01-26 Customized data delivery and network configuration via aggregation of device attributes.
- [20] Soma Bandyopadhyay, Abhijan Bhattacharyya “Generic Middleware Architecture Supporting Heterogeneous Sensors Management for Any Smart System” Necom 2012, July 13 - 15, 2012 Chennai, India.

Authors

Soma Bandyopadhyay has more than 14 years of industry experience in the area of Embedded Systems, Digital Signal Processor, Protocol, Wireless Communications and Ubiquitous Computing. Since 2003 has been associated with Innovation Lab of TATA Consultancy Services (TCS) as senior scientist. Presently her prime focus area is ubiquitous and sensor network and computation. At present she is leading the research and development activity in interoperability, adaptability and QoS aspects of ubiquitous computing. Academically she is an M.Tech & B.Tech in Computer Science & Engineering from the University of Calcutta, India, and graduated in Physics from the same university.



Abhijan Bhattacharyya is presently working as a scientist in Tata Consultancy Services Innovation Lab, Kolkata, India. He has done his Bachelor of Technology in Information Technology from the University of Calcutta, India and Bachelor of Science with Honours in Electronics from the same university. His primary areas of interest are network protocols, wireless baseband communication protocols, digital signal processing, etc. He has a vast industrial experience of working with contemporary wireless protocol layers. His present area of research interest is application layer protocols for constrained devices.

