Hindawi Publishing Corporation
Chinese Journal of Engineering
Volume 2014, Article ID 701829, 8 pages
http://dx.doi.org/10.1155/2014/701829

Research Article

Architecting Service Based Sensor Networks for the Intelligent Assimilation

R. S. Ponmagal

Computer Science and Engineering, Dr. M. G. R. Educational and Research Institute University, Chennai, India

Correspondence should be addressed to R. S. Ponmagal; rsponmagal@gmail.com

Received 8 November 2013; Accepted 15 December 2013; Published 9 January 2014

Academic Editors: H. Lu and I. Smith

The aim of this paper is to propose an architectural model for assimilating distributed sensor networks through cloud paradigm. This strategy can be applied to monitor and control the physical parameters such as temperature, pressure, and level. It is proposed to consider the use of service oriented architecture to program and deploy the sensed parameters. The service oriented architecture for sensor network has been implemented in such a way that, for every specific requirement of the monitor center, the assimilation agent invokes the services of the sensors through a registry and the specific changes in the sensed parameters are also notified as auditable event using push interaction pattern of SOA. The assimilation agent serves as an intelligent component by providing authentication services. This SOA is extended to integrate different types of sensor networks through cloud environment. Hence several sensors can be networked together to monitor different process parameters and they have been assimilated with Internet by registering them as services, hence a complete distributed assimilation environment is exploited.

1. Introduction

The industrial process parameters such as temperature, pressure, and level, are time critical and they have been traditionally monitored at the control centers. The physical parameters measured by analog or digital instruments are converted into electrical parameters such as voltage and/or current, so that it is easier to transmit them to a distant control center. The above methods of monitoring, analysis, and control are possible only within the industrial coverage area. Recent advances in microsensor technology have led to the large-scale deployment of low cost and low power sensing devices to sense the process parameters with computational and wireless communication capabilities. The heterogeneous sensors to sense the different process parameters sensors to sense the different process parameters are networked together to form a distributed sensor network. For such applications, sensor networks could not operate as standalone networks. There must be an effective way for user to gain access to the data produced by the sensor networks. The integration of sensor networks with Internet is one of the open research issues in wireless sensor networks. At the same time, it is envisioned that sensor networks are to be pervasive and ubiquitous. It makes a lot of sense to integrate sensor networks with Internet [1]. It is not possible to give an IP address to every sensor node for their large numbers, thereby making it impossible to directly connect sensor networks with Internet by TCP/IP.

Popularity of cloud computing is increasing day by day in distributed computing environment. There is a growing trend of using cloud environments for storage and data processing needs. Cloud computing provides applications, platforms, and infrastructure over the Internet. It is a new era of referring to access shared computing resources. On the other hand, wireless sensor networks have been seen as one of the most essential technologies for the 21st century where distributed spatially connected sensor nodes automatically forms a network for data transmission. The proposed work provides the solution of integrating the wireless sensor networks with cloud through SOA paradigm, with improved performance. Service oriented architecture is an architectural paradigm and discipline that may be used to build infrastructures enabling consumers and providers to interact via services across disparate domains of technology and ownership.

In the proposed work, an integrated solution with application server (providers), integration agent (through which consumers contact), and a register agent (registry) is designed. The component integration agent acts intelligent by playing the role of authenticating users, processing queries, and disseminating the related data.

2. State of the Art

Current improvements in sensor networking for various applications highlighted the requirement for a reliable integration of sensor networks with the Internet [2]. Data collected by sensor networks are required to be transmitted promptly to users of the Internet for analysis and intelligence gathering [2]. The collected data propagate back to a sink or gateway. The gateway is a more complicated sensor node that has sufficient capabilities to query and communicate with other sensor nodes within the coverage area. Typically, the gateway is connected to an enterprise network or to the Internet in order to send/receive data to/from deployed WSNs. The entire future Internet would benefit from being regarded as being sensitized and adaptive to facilitate innovation while promoting diversity.

Integration of SOA-enabled Networked Sensors in Enterprise Systems via a web service Infrastructure are transmitted from multiple acquisition sources towards one or more processing points, which may be connected to the external networks. Since sensors monitor a common phenomenon, it is likely that significant redundancy among data generated from different sensors would appear [3]. Such redundancy can be exploited to save transmission energy, throughout filtering and data aggregation procedures in the network. Cloud computing has emerged as a promising area to deal with collaborative data and services. It could be a better solution for WSN applications for sharing data and service for large-scale environment. In the present investigation, SOA based architecture has been proposed to support cloud computing in WSN. The need of WSN integration with cloud is essential for real time processing of heterogeneous data sources in order to make critical decisions. Invocation of services on the cloud one after another is done to carry out the complex tasks and highly swift data processing using the processing power of the cloud to provide a quick response to the user [4].

A sensor network can be regarded simply as an edge network upon which the Internet needs to make no further demands; as long as the boundary router can respond to IP traffic, the sensor network is free to use whatever techniques within itself that it wishes [5]. Rather than constructing networks, topologies, protocols, and adaptive approaches, this work proposes an extensible internet working architectures through SOA for enterprise services and cloud. It is possible to classify the integration approaches between the Internet and wireless sensor networks in two different ways: stack-based and topology-based classification. In the stack-based classification, the level of integration between the Internet and a WSN depends on the similarities between their network stacks. A WSN can be completely independent of the Internet (Front-End), be able to exchange information with Internet hosts (Gateway), or share a compatible network-layer protocol (TCP/IP). On the other hand, in the topology-based classification the level of integration depends on the actual location of the nodes that provides access to the Internet [6].

3. Proposed Architecture

The SOA is a prominent approach for developing distributed system that delivers application functionality as services to either end-user applications or other services. Web services are technology that is well suited for implementing a service-oriented architecture and delivering business logic as services that can be published, discovered, and invoked over the coupled application components using any programming language, any protocol, or any platform.

3.1. Customised Layered Architecture. The customized layer architecture of integrating wireless sensor networks with SOA for enterprise service such as cloud is shown in Figure 1. Wireless sensor networks can exchange data and require business logic before communicating back to enterprise system. Enterprise services have the ability to impact the business requirements directly and technical architecture of combining WSN and the Internet.

The level 0 consists of clusters of sensor nodes which provide physical access to the environment. Depending on the parameters requirement, particular type of sensors is attached as clusters. In the proposed architecture, communication and routing are based on cluster mechanisms. Sensor nodes form clusters. Each cluster node is treated as base station. A head of each cluster is selected according to some negotiated rules. The powerful node acts as cluster head node, other nodes sense the data and forward them to the cluster nodes.

Cluster nodes have the capability to identify the locations of sensor nodes and obtaining the sensor data from a sensor at preferred location. Cluster head nodes are connected to web server with the help of sensor interface where web service for reading the sensor data is stored. The duty of level 1 provides appropriate abstraction and functionality for interfacing between upper layer (level 3) and lower layer (level 0). The responsibility of sensor interface is looking beyond collecting the information from the cluster head nodes and storing it in web server. It also provides facility to inject the services into cluster head nodes according to the service requirement. Level 1 is not unique to all sensor networks. Research and development work has been carried out for developing unique middleware for heterogeneous sensor networks. Level 2 provides services to users. Security, server, database, and communication are the five components that make up the service layer. Level 3 is called application layer that provides facilities to access the system application. QoS mechanisms define the QoS constrains of the system [7]. New middleware is proposed that includes some components that are available in another good middleware. These components provide the advancement and improved computation in service oriented architecture to take care of the processing load and to deal with the real time demand of query processing. It allows the seamless integration and execution of process logic for sensor

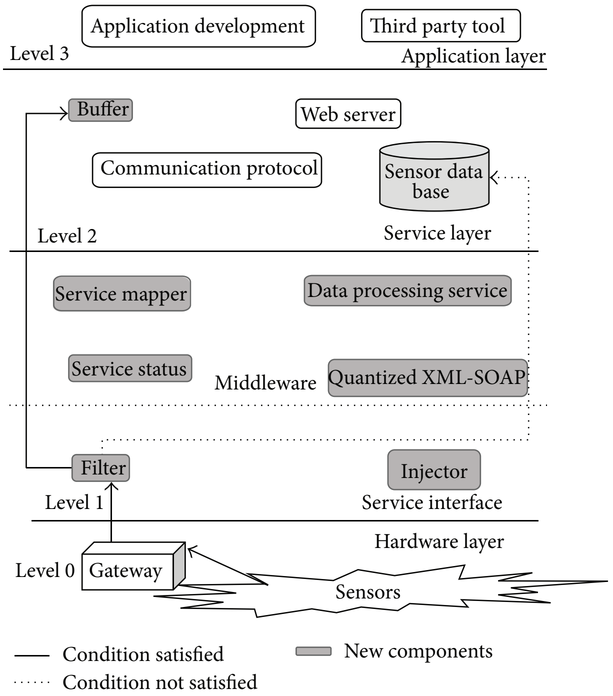


FIGURE 1: Customized Layer architecture.

networks. The new components included in the middleware are given below.

Quantized XMLSOAP. It contains the quantization technique which is used to minimize the data which has to be sent by the sensor nodes. This technique depends on the past transmission records. A comparison is performed on the present data to be sent and the previously transmitted data. It is in XML form. Hence, communication overhead is reduced and significant savings in the bandwidth are ensured. It is communicated in the form of compressed SOA messages.

Filter Box. The gateway passes the data to the filter box, judges the sensed data, and declares it as normal or abnormal. Normal data is forwarded to the database, and abnormal data moves into the buffer for further examination if required later.

Sensor Data Processing Services. It is a preconfigured business process service and provides business context to sensor network events. These components are provided with business logic.

Service Mapper. Where and which service should be deployed? This component is provided with the description of deployment using agents. There are three agents, namely, Requestor Agent (RA), Registry Service Agent (RS), and Authentication Agent (AA), which are defined based on the functionalities such as service availability, description, selection, and authentication.

RA Acts as a Service Requestor. It can help the user to register some services to RS and get authentication from AA. RA is a program for user, and it can update the service definition from RS. It also extracts the information from AA to a user.

RS is a Service Registry. It is responsible for the service register. It can broadcast the service to RA, collect different service descriptions, and get the related authentication information from AA. AA plays a role of authenticating a user and responding to the user.

Service Status. This component indicates the delivery status of the system.

3.2. Service Oriented Architecture. Figure 2 depicts the service oriented architecture model for distributed networked industrial sensors. The sensor services are deployed into the application server as service description which has the location information and provides a service end point, a target namespace, and a transport name. This component makes use of message exchanges in the XML data format. Since programming sensor networks remains too complex with existing programming languages and techniques, using XML messages [8] in sensor networks will optimize the way of getting information from the network.

To facilitate orchestration and aggregation of services into processes and applications, an eb-XML-registry-repository (register agent) is used. The registry-repository provides a single view of all services. The sensor services are published into the eb-XML registry using wsdl. The lists of services are discovered and invoked by the sensor applications (client), using SOAP with Quantized-XML messages. The client communications are passed through the integration agent. The AA also takes care of the authentication of the users and delivering the required parameters using push interaction pattern. This pattern can be triggered by multitude of events, here an auditable event, triggering (when the process parameters exceed some threshold) the message sent to the client.

4. Implementation

Here it is described how the sensed parameters are converted into web service and how they have been deployed. Web services are application components that are designed to support interoperable machine-to-machine interaction over a network. WSNs can be classified into two types on the basis of their mode of operation or functionality and the type of target application: proactive networks and reactive networks. In proactive networks, the nodes periodically switch on their sensors and transmitters, sense the environment, and transmit the data of interest. Thus, the applications requiring periodic data monitoring are suited for this kind of networks. In reactive networks, the nodes react immediately to sudden and drastic changes in the value of a sensed attribute. The time-critical applications are suited for it. To deal with the different applications, two kinds of solution are designed: pull process for proactive application and push process for reactive application. In integration agent, some components are responsible for differentiating those application and are allocated different tasks.

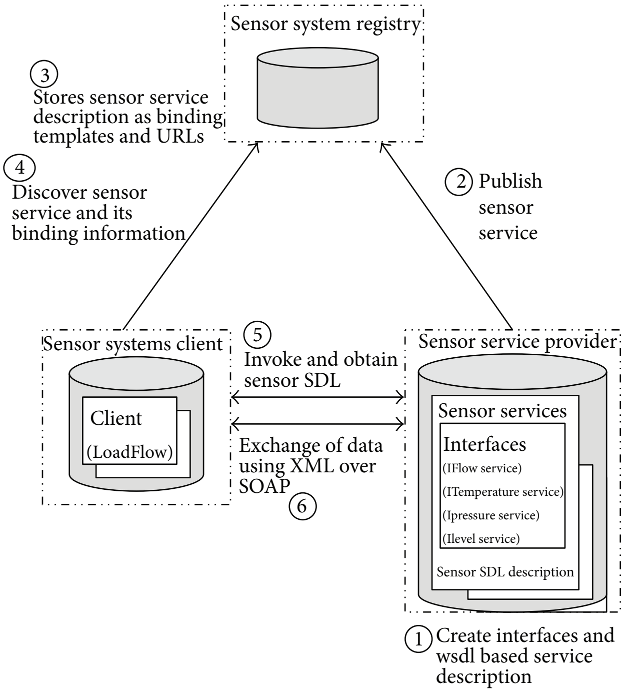


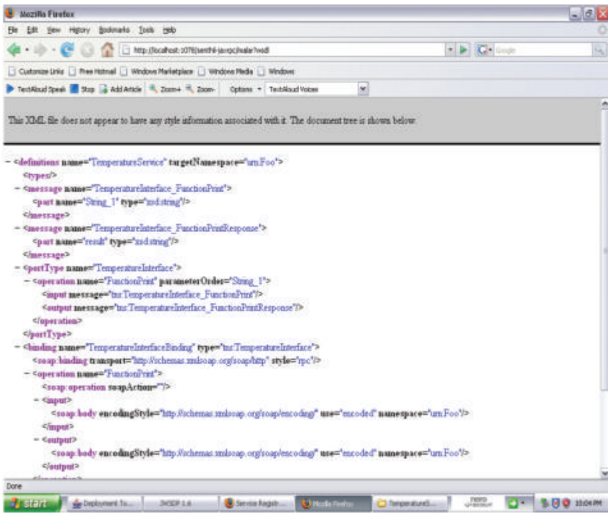
Figure 2: Service oriented architecture for sensor networks.

4.1. Simulation of Sensors. The sensor network is built as prototype, using random numbers generated as sensed parameters. In a sensor network, this provides a number of advantages, including shielding the user from faulty sensors and reducing the number of expensive sensor readings and radio transmissions that the network must perform. This general architecture acts as the proper platform for answering queries and interpreting data from real world environments like industrial sensornets, as conventional database technology is poorly equipped to deal with lossiness, noise, and nonuniformity inherent in such environments.

4.2. Sensors as Service Deployment. Here, it is described how the sensed parameters are converted into web service and how they have been deployed. Web services are application components that are designed to support interoperable machine-to-machine interaction over a network. This interoperability is gained through a set of XML-based open standards, such as the Web Services Description Language (WSDL), the Simple Object Access Protocol (SOAP), and Universal Description, Discovery, and Integration (UDDI). These standards provide a common and interoperable approach for defining, publishing, and using web services.

J2EE 1.4 SunAppserver is used as service provider. The J2EE 1.4 platform provides comprehensive support for web services through the JAX-RPC 1.1 API, which can be used to develop service endpoints based on SOAP.

The interface and implementation files for the process parameters such as temperature, pressure, and flow are written. Configuration files are written to specify the XML [4] namespace and target namespace. These files are compiled to generate WSDL (contains possible inputs and server's address) for client reference and mapping file (port number and service endpoint location) for server refernce. With deployment tool war files are generated for the services written and deployed into the server. Figure 3 shows the deployed temperature service at service provider after following the above steps.


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:1078/senthil-jaxrpc/malar?wsdl
Customize Links Free Hotmail Windows Marketplace Windows Media Windows
TextAloud Speak Stop Add Article Zoom+ Zoom- Options TextAloud Voices
Done
start Deployment To... JWSDP 1.6 Service Regis... Mozilla Firefox TemperatureS...
10:04 PM

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <definitions name="TemperatureService" targetNamespace="urn:Foo">
    <types/>
  - <message name="TemperatureInterface_FunctionPrint">
      <part name="String_1" type="xsd:string"/>
    </message>
  - <message name="TemperatureInterface_FunctionPrintResponse">
      <part name="result" type="xsd:string"/>
    </message>
  - <portType name="TemperatureInterface">
    - <operation name="FunctionPrint" parameterOrder="String_1">
        <input message="tns:TemperatureInterface_FunctionPrint"/>
        <output message="tns:TemperatureInterface_FunctionPrintResponse"/>
      </operation>
    </portType>
  - <binding name="TemperatureInterfaceBinding" type="tns:TemperatureInterface">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
    - <operation name="FunctionPrint">
        <soap:operation soapAction=""/>
      - <input>
          <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="urn:Foo"/>
        </input>
      - <output>
          <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded" namespace="urn:Foo"/>
        </output>
```

Figure 3: Temperature sensor as service.

4.3. Quantised XML-SOAP. Sensor will receive the request from the sink and it will give the sensed information as the response to the intended user. The response which has been sent by the sensors has to be taken into account, because the data transmission rate affects the power consumption in the sensor node. So the size of the data has to be reduced. The data capacity reduction will reduce the space utilized, time for data transmission, and the energy consumption. This will lead to the increase in the lifetime of the sensor nodes. The quantization is a technique which is used to minimize the data which has to be sent by the sensor nodes. This technique depends on the past transmission records. A comparison is performed on the present data to be sent and the previously transmitted data. Depending on the variations the quantization [9] is performed on the data and then the data is transmitted. The need of this system is to save time, space, and energy, to increase the flexibility, to manage the power consumption, and to make the sensor data in the suitable format for internet accessing. The advantages of this system are memory efficiency, runtime efficiency and processability. Memory efficiency is attained by representing high amount of data with a low amount of allocated memory. Runtime efficiency is achieved through minimal number of processing cycles for XML data. And then the power consumption is reduced to a large extent which will increase the lifetime of the sensor nodes. As the data transmitted by the sensor increases the battery power will get lowered. This will affect the lifetime of the sensor. To reduce the data rate the quantization technique is adopted in WSN. Quantization in the sensor node is the process of converting large data into reduced data. The process is initiated by comparing the actual data and the last recently transmitted data. Let A be the sensor node. The data transmitted from A at the time t is M;

the size of the data is 4 kb. Let B be the data transmitted at the time $t + 1$ from A, whose size is 6 kb, the power consumed for transmitting M is 3 mw and N is 4 mw. As the data rate increases the power consumed also increases. By introducing the quantization method the existing scenario can be changed.

Without quantization

Data transmitted by A at time $t = M$ (1)

Data transmitted by A at time $t + 1 = N$ (2)

where $N > M$

Power consumption of A at time $t = P1$ (3)

Power consumption of A at time $t + 1 = P2$ (4)

where $P2 > P1$

After introducing quantization

Data rate:

Data sent by A at time $t = M$ (5)

Data has to be sent by A at time $t + 1 = N$ (6)

Actual data sent by A at time $t + 1 = N - M$ (7)

$= N'$

where $N' < M$

Power consumption:

Power consumed to transmit M at time $t = P1$ (8)

Power consumed to transmit M at time $t + 1 = P2$ (9)

Actual power consumed to transmit M

at time $t + 1 = P2 - P1$ (10)

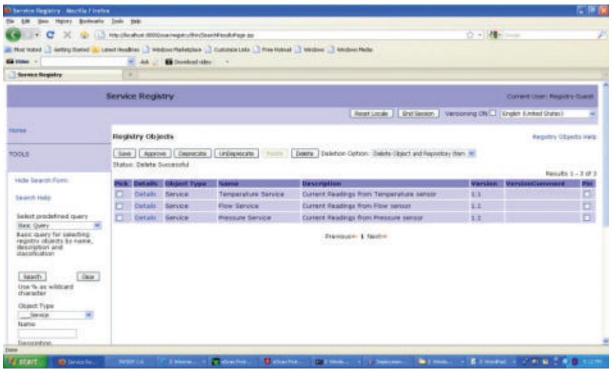
$= P2'$ where $P2' < P1$

The client sends a request to the sensor node. The request is processed and a quantized response is produced. The quantized response is obtained through the differential value, that is, the difference between the current data to be sent and the last recently sent data. The quantized response is mapped to a Q-XML template in the server and then the actual response is sent to the client.

4.4. Sensor Registry. To facilitate orchestration and aggregation of services into processes and applications, registry (register agent) is used. To publish the services, the eb-XML registry available with tomcat50-java web services developer package is used. The WSDL files for sensor services are used and the appropriate service bindings are set to register the services at tomcat server. Figure 4 depicts the registered services at the registry.

5. Cloud Architecture

This section outlines the architecture that enables us to couple the services provided by remote sensor networks over the Internet. The cloud computing supports the interaction of consumers and WSN through a standard interface. The increasing deployment of wireless communication technologies has made it possible to use online storage for ordinary data storage in pervasive computing environments. At the same time, the emerging cloud storage solutions which can significantly reduce the managerial costs and offer flexible reliable storage have shown their strong abilities to meet the growth storage requirements in pervasive computing environments such as WSN. The popular cloud platform services are running mostly at the operating system and programming language level rather than at the level of the SOA platform. To be really useful for SOA, cloud platforms [10] should include enterprise service buses, service registries, and other SOA platform components. For integration between multiple clouds and on-premise applications, a clean SOA architecture simply eliminates the internal integration fur-ball spreading in the clouds. Since wireless sensor networks are limited in their processing power, battery life, and communication speed, cloud computing usually offers the opposite, which makes it attractive for long term observations for shared use of data and services in different kinds of project analysis. The huge amount of data, which a sensor network is able to deliver, demands a powerful and scalable storage and processing infrastructure. Today, wireless sensor network platforms that perform sensing and complex calculations are most of the time constrained in their capabilities. One appropriate way to solve this issue is offline processing of sensor data if the resources are not sufficient. The combination of wireless sensor networks, with their huge amount of gathered sensor data and their limited processing power, with a cloud computing infrastructure makes it attractive in terms of (i) integration of sensor network platforms from different vendors, (ii) scalability of data storage, (iii) scalability of processing power for different kinds of analysis, (iv) worldwide access to the processing and storage infrastructure, (v) resource optimization, (vi) being able to share the results more easily, (vii) and using pricing as one more criteria for the IT infrastructure.


Service Registry

Current User: Registry Guest

Registry Objects

Status: Delete Successful

Results 1 - 3 of 3

Pick	Details	Object Type	Name	Description	Version	VersionComment	Pin
□	Details	Service	Temperature Service	Current Readings from Temperature sensor	1.1		□
□	Details	Service	Flow Service	Current Readings from Flow sensor	1.1		□
□	Details	Service	Pressure Service	Current Readings from Pressure sensor	1.1		□

Select predefined query: Basic Query

Basic query for selecting registry objects by name, description and classification

Use % as wildcard character

Object Type: Service

Figure 4: Registered temperature sensors as service in the sensor registry.

The main issue in WSNs is to deliver the collected information efficiently with minimum delays to data centers, where different pieces of information are brought together to provide a better environment. This can be achieved by integrating the WSNs and the Internet with existing Quality of Service (QoS) techniques. The integration of WSNs and the Internet is becoming more and more important because of the numerous numbers of WSNs that will join the Internet

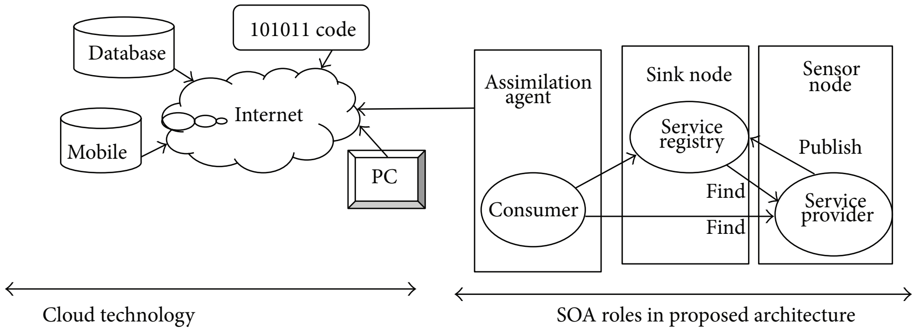


Figure 5: Integrated sensors as service through cloud.

domain. Currently, the data gathered by WSNs are delivered to data centers with best-effort services, which means that delay-sensitive and time-sensitive data are subject to be dropped or delayed in congested networks. The integration of WSN with cloud computing provides quick response to the users by using the immense processing power of the cloud. The vast amount of sensed data can be processed, analysed, and stored using computational and storage resource of the cloud. Cloud computing allows sharing of sensor resources by different users and applications under flexible scenario. Sensor resources do not have direct connection with cloud. Hence, a framework is necessary to manage the data from sensor network and take it over long environment. In this chapter, SOA architecture is extended to cloud by using assimilation agent which acts as a bridge between the two technologies. This extended architecture is shown in Figure 5.

Google App Engine and Microsoft Windows Azure are two competitors in the cloud computing business. Both provide on-demand compute and storage capabilities to host, scale, and manage Web applications and services on the Internet hosted in a datacenter. The preferred communication method from the client's computer to the cloud computing environment is based on web services, which is a proven technology and supports interoperability. The use of web services brings a lot of flexibility for the development of those clients, which are hosted in the customer's environment so that the customer's legacy systems can be easily integrated.

5.1. Handling Quantized XML Form of Data from Sink to User End. A bridge program is responsible to read the sensed data from simulator and convert into XML form of data and store it on Web Server. The following code shows that sample XML code of sensed data:

```
<?XML version="1.0"?>

<Units>

<Level><SensorID>LS1P1
</SensorID><SensorTime>17:03:47
</SensorTime><SensorValue>null
</SensorValue></Level>

<Pressure><SensorID>PS1P1
</SensorID><SensorTime>09:28:02
</SensorTime><SensorValue>1
</SensorValue></Pressure>

<Level><SensorID>LS1P1
</SensorID><SensorTime>17:02:46
</SensorTime><SensorValue>null
</SensorValue></Level>

</Units>
```

5.2. Deployment of Sensed Data into the Client End through STAX. STAX is Elastic Java application platform for EC2. The fastest way for Java EE developers to build, manage, and scale applications in the cloud. Stax provides easy deployment of test and production environments, a local development model, and strong integration with existing development tools, frameworks, and processes.

```
buildXML(strFileName,strCurrentReading,out);

out.println(strCurrentReading);

/* Batch File Running*/String command = "cmd/C
start D:/tomcat50-

jwsdp/webapps/servlets-examples/WEB-
INF/classes/cloud.bat";

Runtime rt = Runtime.getRuntime();

rt.exec(command);}

else{ out.println("Error over File making");}
```

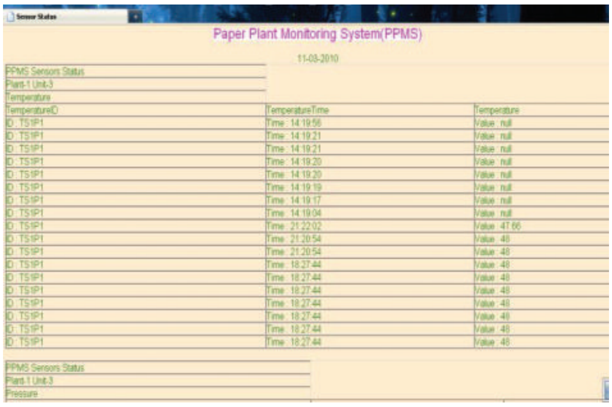
The integration agent is responsible to read the sensed data from sink and deploy into the client end by using STAX. The Paper Plant Monitoring system is shown in Figure 6 where the sensed process parameter is deployed in STAX cloud.

6. Experimental Analysis

6.1. Web Service Message Sizes for Different Encapsulations. The web services implementation is centered around the WSDL standard. WSDL supports several encapsulation protocols for sending the method calls to the webservice host device. Currently supported encapsulation protocols are SOAP, HTTP, and MIME [11]. The binding section of the WSDL file describes which one of these standards is used for accessing device methods. Currently, SOAP is considered the de-facto standard for transferring web service messages.

Table 1: Web service message sizes for different encapsulations.

Method name	SOAP 1.1	SOAP 1.2	HTTP	QXML SOAP
GetTemperature	491	442	162	420
GetTemperatureResponse	479	499	258	450



Paper Plant Monitoring System(PPMS)

11-03-2010

PPMS Sensors Status
Plant-1 Unit-3
Temperature

TemperatureID	TemperatureTime	Temperature
ID : TS1P1	Time : 14:19:56	Value : null
ID : TS1P1	Time : 14:19:21	Value : null
ID : TS1P1	Time : 14:19:21	Value : null
ID : TS1P1	Time : 14:19:20	Value : null
ID : TS1P1	Time : 14:19:20	Value : null
ID : TS1P1	Time : 14:19:19	Value : null
ID : TS1P1	Time : 14:19:17	Value : null
ID : TS1P1	Time : 14:19:04	Value : null
ID : TS1P1	Time : 21:22:02	Value : 47.66
ID : TS1P1	Time : 21:20:54	Value : 48
ID : TS1P1	Time : 21:20:54	Value : 48
ID : TS1P1	Time : 18:27:44	Value : 48
ID : TS1P1	Time : 18:27:44	Value : 48
ID : TS1P1	Time : 18:27:44	Value : 48
ID : TS1P1	Time : 18:27:44	Value : 48
ID : TS1P1	Time : 18:27:44	Value : 48
ID : TS1P1	Time : 18:27:44	Value : 48
ID : TS1P1	Time : 18:27:44	Value : 48

PPMS Sensors Status
Plant-1 Unit-3
Pressure

Figure 6: Temperature sensor as a service in cloud.

However, a fully standard compliant SOAP server implementation requires considerable computational and memory resources than those provided by low-power processors such as MSP430. The SOAP envelop is very verbose and also needs significant radio bandwidth. The web service message sizes for different encapsulations are shown in Table 1. The proposed Q Xml-SOAP message encapsulation, which is obtained from the temperature sensor simulations, occupies less size compared to the previous SOAP encapsulations, which are taken from the previous research. This is due to the fact that the proposed Qxmlised messages are well compressed SOA messages. When this message is transferred over the wired or wireless link it occupies only less bandwidth; hence the performance is improved.

6.2. Access Time for Various Resources. The resource access time for various numbers of resources with SOA environment and cloud environment are recorded and graphically represented in Figure 7.

From Figure 7, it is noted that, since the availability of resource is rich in cloud, the cloud architecture takes less access time to locate the service in the Internet compared to SOA architecture. Access time is defined as the time that takes to search and locate the required resources for the service in cloud. Moreover it is observed that the access time becomes linear when the number of resources requirements is increased in the cloud; hence it is possible to access more number of resources with cloud environment, which shows the performance improvement in terms of scalability. The scalability of this approach seems to be unlimited, since wireless sensor networks operate independently and are connected to the cloud computing environment through a scalable number of wireless sensor networks. The cloud computing environment itself offers a scalable infrastructure, which makes it very attractive.

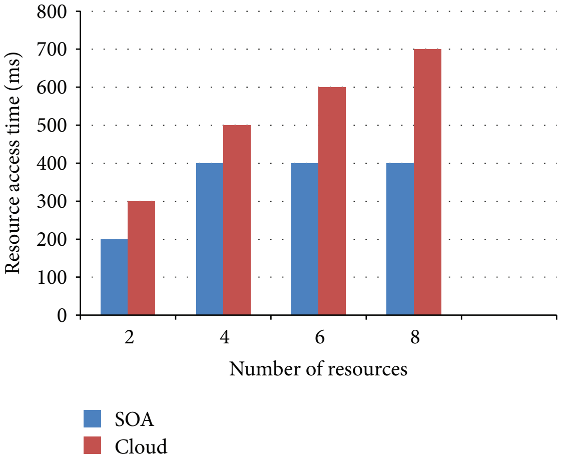


Figure 7: Resource access time SOA versus cloud.

7. Conclusion

The proposed paper aimed at working on applying and extending the service oriented paradigm to sensor network application engineering such as distributed manufacturing; we derived the requirements for the sharing of sensor networks as new resources in this domain. The necessary abstraction was implemented using the service oriented process parameters with performance guarantees, which lead to the intelligence integration into the Internet. This solution is extended to sensor clouds offering Sensor as a Service (SeaaS). This could also be extended to serve as an adaptive web information system of sensor services which takes into account the user preferences and service preferences. The WSN data is distributed over Internet using cloud computing technology. The basic properties of cloud like interoperability, reliability, availability, and security are achieved which leads to the intelligent assimilation of heterogeneous sensor networks. These proposed models provide fast delivery of data to client machine. Failure time recovery can take place instantly. A potential future work is to deploy this proposed work in Industry in real time and test it.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Shi and W. Liu, “A service-oriented model for wireless sensor networks with internet,” in *Proceedings of the 5th International Conference on Computer and Information Technology (CIT ’05)*, pp. 1045–1049, September 2005.

[2] S. Weilian and A. Bassam, “QoS integration of the internet and wireless sensor networks,” *WSEAS Transactions on Computers*, vol. 7, no. 4, pp. 121–135, 2008.

[3] F. C. Delicato, P. F. Pires, L. Pirmez, and L. F. R. da Costa Carmo, "A service approach for architecting application independent wireless sensor networks," *Cluster Computing*, vol. 8, no. 2-3, pp. 211–221, 2005.

[4] H. Lim, "Sensor Cloud: Towards Sensor-Enabled Cloud Services," Intellysys'09 NTU, 2009.

[5] Y. Dogan and D. Adam, "Efficient application integration in IP-based sensor networks," in *Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys '09)*, pp. 978–991, ACM, Berkeley, Calif, USA, 2009.

[6] R. Roman and J. Lopez, "Integrating wireless sensor networks and the internet: a security analysis," *Internet Research*, vol. 19, no. 2, pp. 246–259, 2009.

[7] M. Ceriotti, "Quality of service in wireless sensor networks: a survey," *Sensors Journal*, vol. 9, no. 8, pp. 139–150, 2008.

[8] N. Hoeller, C. Reinke, J. Neumann, S. Groppe, D. Boeckmann, and V. Linnemann, "Efficient XML ge within Wireless Sensor Networks," in *Proceedings of the 4th Annual International Conference on Wireless Internet (WICON '08)*, pp. 17–19, Maui, Hawaii, USA, November 2008.

[9] A. Rosenthal, P. Mork, M. H. Li, J. Stanford, D. Koester, and P. Reynolds, "Cloud computing: a new business paradigm for biomedical information sharing," *Journal of Biomedical Informatics*, vol. 43, no. 2, pp. 342–353, 2010.

[10] B. Nissanka, K. Aman, G. Michel, and Z. Feng, "Tiny web services:design and implementation of interoperable and evolvable sensor networks," in *Proceedings of the 6th ACM conference on Embedded Network Sensor Systems (SenSys '08)*, pp. 253–266, 2008.

[11] R. S. Ponmagal and J. Raja, "An Extensible cloud Architecture model for heterogeneous sensor services," *International Journal of Computer Science and Information Security*, vol. 9, no. 1, pp. 147–155, 2011.

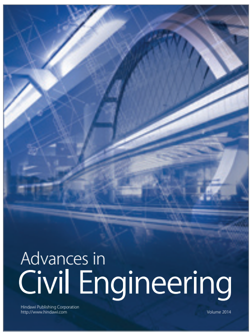
Journal of
Engineering

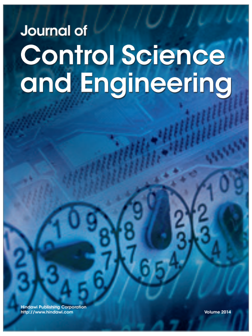
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

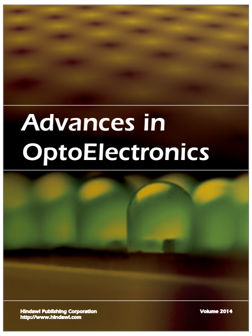
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

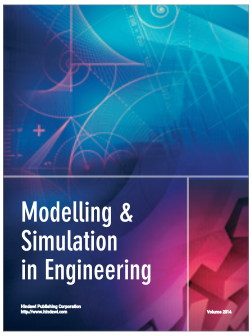
Journal of
Robotics

Journal of
Electrical and Computer Engineering

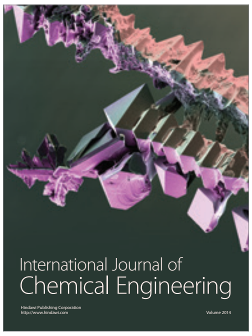

Advances in
OptoElectronics

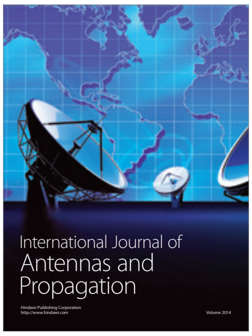
VLSI Design

International Journal of
Navigation and Observation

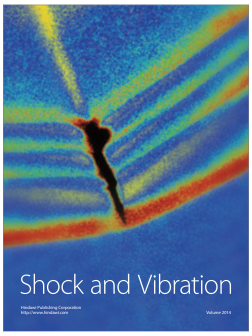

Modelling & Simulation in Engineering

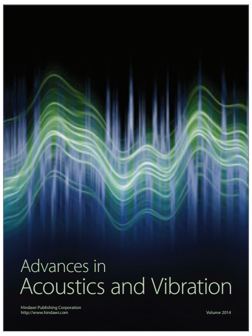
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration