

Architecting Service based Sensor Networks for the adept assimilation

R.S.Ponmagal^{#1}

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

Abstract— 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 invoke 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 adept 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.

Keywords— Sensor, Service, SOA, Internet, assimilation, cloud.

I. 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 method of monitoring, analysis and control are possible only within the industrial coverage area. Recent advances in micro sensor 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 are networked together to form a distributed sensor network. For such applications, sensor networks could not operate as stand-alone networks; there must be an effective way for user to gain access to the data produced by the sensor networks. The assimilation of Sensor Networks with Internet is one of the open research issue 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.

The proposed work provides the solution of assimilating the sensor networks with Internet through SOA paradigm, with improved performance. Service Oriented Architecture is an architectural paradigm and discipline that may be used to build infrastructures enabling those with needs (consumers) and those with capabilities (providers) to interact via services across disparate domains of technology and ownership. In the proposed work, an integrated solution with application server (providers), assimilation agent (through which consumers contact), and a register agent (registry), is designed. The component assimilation agent acts intelligent by playing the role of authenticating users, processing queries, and disseminating the related data.

II. STATE OF 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. Sensor information is 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 are real time processing of heterogeneous data sources in order to make critical decisions, invocation of services on the cloud one after another 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]. The integration of cloud provides sharing of sensor resources by different users and applications under flexible usage scenario.

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]. In this view, the future internet need makes no statements about, or provisions for, sensor networks or indeed any other special-purpose networks that may be connected. Identifying the core adaptive structures are a further step towards this goal. Rather than construct networks, topologies, protocols and adaptive approaches, this work proposes an extensible internetworking architectures through SOA for enterprise services and Cloud. In the topology-based classification of WSN the level of integration depends on the actual location of the nodes that provides access to the Internet [6].

III. PROPOSED ARCHITECTURE

Web services are a technology that is well suited for implementing a service-oriented architecture and deliver 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. 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 requires 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.

A. Customisation

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 act as cluster head node, other nodes sense the data and forward 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 stored 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

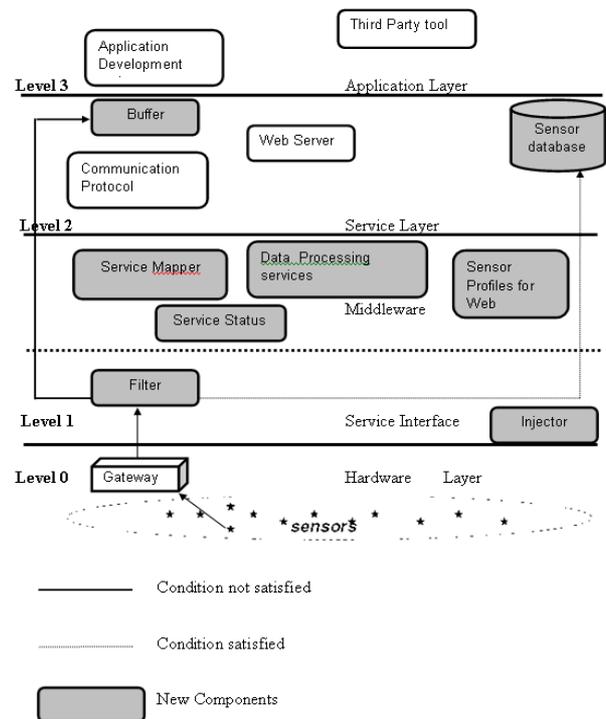


Fig. 1 Customized layered architecture

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

Sensor Profile for Web Service (SPWS): It contains the sensor template and service template as a message in the formal specification. It is in XML form. Hence, communication overhead is reduced and significant savings in the bandwidth is ensured. It is communicated in the form 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 pre-configured business process service and provide 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.

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

B. Architecture

Fig 2 depicts the SOA 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.

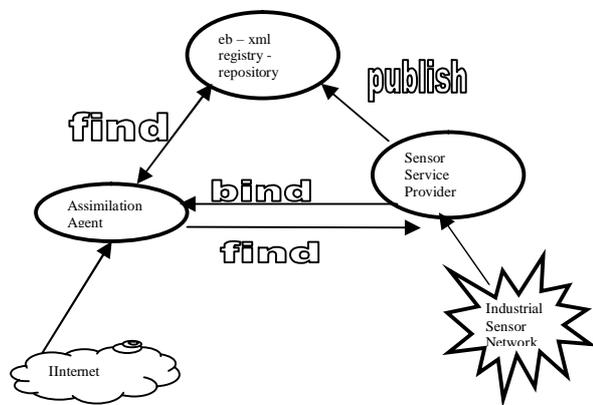


Fig. 2 SOA model for Sensor Networks

The lists of services are discovered and invoked by the sensor applications (client), using XML-RPC and SOAP messages. The client communications are passed through the Integration Controller. The IC 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, trigger (when the process parameters exceeds some threshold) the message sent to the client.

IV. 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. 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. In proactive sensor 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 sensor 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.

A. 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 non-uniformity inherent in such environments.

B. Sensors as service deployment

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 reference. With deployment tool war files are generated for the services written and deployed into the server. Fig 3 shows the deployed temperature service at service provider after following the above steps.

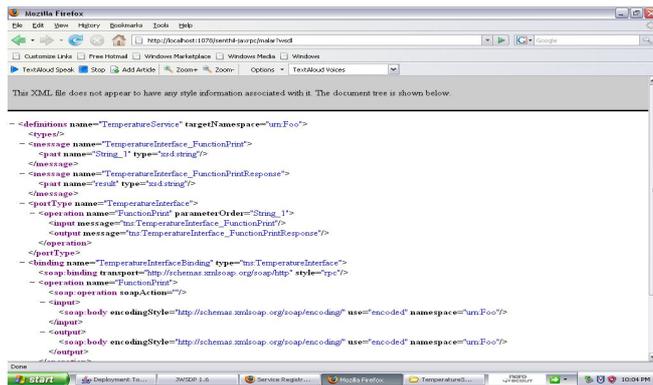


Fig. 3 Temperature Sensor as a Service

C. Quantized 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 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. 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 a large data into a 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 be 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 be 3 mw and N be 4 mw. As the data rate increases the power consumed also increases. By introducing the quantization method the existing scenario can be changed.

Data transmitted by A at time t = M
 Data transmitted by A at time t+1 = N

where $N > M$
 Power consumption of A at time t = P1
 Power consumption of A at time t+1 = P2
 where $P2 > P1$

After introducing quantization

Data rate :
 Data sent by A at time t = M
 Data has to be sent by A at time t+1 = N
 Actual data sent by A at time t+1 = N-M
 = N'
 where $N' < M$

Powerconsumtion :

Power consumed to transmit M at time t = P1
 Power consumed to transmit M at time t+1 = P2
 Actual power consumed to transmit M at time t+1 = P2 - P1
 = 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, i.e. 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.

D. 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. Fig 3 depicts the registered services at the registry.

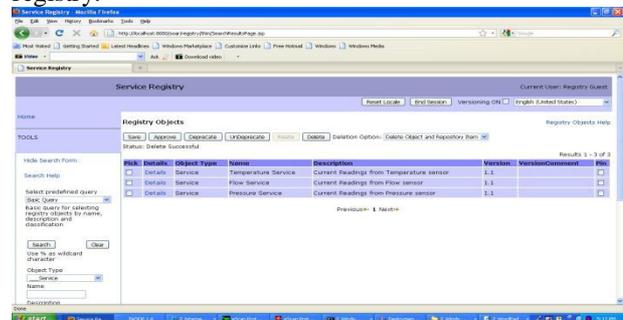


Figure 4. Registered Temperature Sensor as a Service

V. 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.

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) be able to share the results more easily, vii) and using pricing as one more criteria for the IT infrastructure.

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 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 [10] 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 section, SOA architecture is extended to cloud by using Assimilation agent which acts as a bridge between the two technologies.

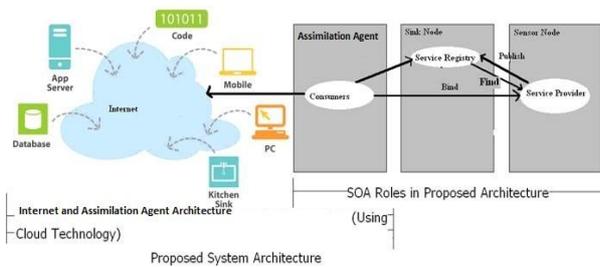


Fig 5. Assimilated sensors as a Service through Cloud

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.

A. Handling Quantized Xml form of Data from Sink To User End

A bridge program is responsible to read and 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>
```

B. Deployment of Sensed Data Into The Client End By 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 Assimilation Agent is responsible to read the sensed data from Sink and deploy into the client end by using STAX. The following snapshot shows the deployed sensed data available to client end through STAX cloud.

The screenshot shows a web application titled 'Paper Plant Monitoring System (PPMS)' with a date of 11-03-2010. It displays 'PPMS Sensors Status' for 'Plant-1 Unit-3'. The data is organized into a table with columns for Temperature, TemperatureID, TemperatureTime, and Temperature. The table lists multiple sensor readings for 'D-TS1P1' at various times, with values ranging from null to 48 and 47.66. Below this, there is a section for 'Pressure' with a value of 48.

Fig 6. Temperature sensor as a Service in Cloud

VI. EXPERIMENTAL ANALYSIS

A. Web service messages 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. 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 encapsulations, which are 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.

TABLE I
WEB SERVICE MESSAGE SIZES FOR DIFFERENT ENCAPSULATIONS

Method Name	SOAP1.1	SOAP1.2	HTTP	Q Xml-SOAP
GetTemperature	491	442	162	420
Get Temperature Response	479	499	258	450

B. 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 8

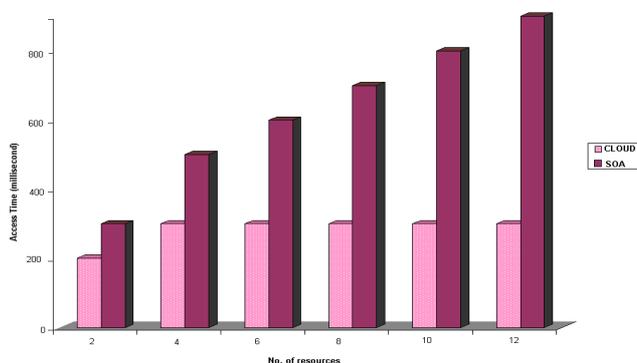


Fig 7. Resource Access time SOA vs Cloud

From the 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

VII. CONCLUSIONS

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 in to 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 adept assimilation of heterogeneous sensor networks. These proposed models provide fast delivery of data to client machine. Failure time recovery can takes place instantly. A potential future work is to deploy this proposed work in Industry in real time and test it

REFERENCES

- [1] Jinglun Shi, Weiping Liu "A Service-oriented Model for Wireless Sensor Networks with Internet" Proceedings of the 2005 The Fifth International Conference on Computer and Information Technology (CIT'05).
- [2] Weilian, S. and Bassam, A., "QoS Integration of the Internet and Wireless Sensor Networks", WSEAS Transactions on Computers, ISSN: 1109-2750, Vol.7, No.4, pp.121-135, 2008.
- [3] Flavia, C., Paulo, F., Luci, P. and Luiz, F. "A Service Approach for Architecting Application Independent Wireless Sensor Networks", Springer Science + Business Media, Inc. Manufactured in The Netherlands, Cluster Computing 8, 211-221, 2005.
- [4] Lim, H. "Sensor Cloud: Towards Sensor-Enabled Cloud Services", Intellysys'09 NTU, 2009.
- [5] Dogan, Y., Adam, D. "Efficient Application Integration in IP-Based Sensor Networks", BuildSys'09, Berkeley, CA, USA, ACM, pp. 978-991,2009.
- [6] Rodrigo, R., Javier, L. and Cristina, A. "Do Wireless Sensor Networks Need to be Completely Integrated into the Internet?", Internet Research: Electronic Networking Applications and Policy, Vol. 19, No. 2, pp.246-259,2010.
- [7] Ceriotti, M. "Quality of Service in Wireless Sensor Networks: a survey", Sensors Journal, Vol 9, No. 8, pp.139-150,2008.
- [8] Nils Hoeller, Christoph Reinke, Jana Neumann, Sven Groppe, Daniel Boeckmann, Volker Linnemann, "Efficient XML Usage within Wireless Sensor Networks" WICON '08, November 17-19, 2008, Maui, Hawaii, USA
- [9] L.Aneesh Euprazia and R.S.Ponmagal, "Energy Efficient Quantized- XML for Wireless Sensor Networks", IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI) (13):31-33, 2011. Published by Foundation of Computer Science.
- [10] Arnon Rosenthal, Peter Mork, Maya Hao Li, Jean Stanford, David

Koester, Patti Reynolds, " Cloud computing: A new business paradigm for biomedical information sharing" ,Journal of Biomedical Informatics 43,432-353,2010.

- [11] Nissanka, B., Aman, K., Michel, G. and Feng, Z. "Tiny Web Services: Design and Implementation of Interoperable and Evolvable Sensor Networks", SenSys'08 pp. 1-14, 2008.
- [12] R.S. Ponmagal, 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, January 2011.
- [13] R.S. Ponmagal, J.Raja, "SAMON – C: Sensor Access and Monitoring in Cloud Environment", CiiT International Journal of Networking and Communication Engineering, Vol.3, No.2, pp.136-144, February2011
- [14] R.S.Ponmagal , "An Efficient and Extensible Service based Sensor Access Architecture", International Journal of Power Control Signal and Computation(IJPCSC) Vol. 4. No.2. pp.117-128 April - June 2012 ISSN: 0976-268X.