Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 5 (3): 352-362, 2013
ISSN 1815-7432 / DOI: 10.3923/rjit.2013.352.362

A Smart Train Using the DPWS-based Sensor Integration

[1]V. Venkatesh, [1]V. Vaithayanathan, [2]Pethuru Raj, [1]K. Gopalan and [1]R. Amirtharajan
[1]SASTRA University, Thanjavur-613001, Tamilnadu, India
[2]Wipro Technologies, Bangalore, 560035, India

Corresponding Author: V. Venkatesh, SASTRA University, Thanjavur-613001, Tamilnadu, India

ABSTRACT

Without an iota of doubt, technology has been instrumental in envisioning and realizing a host of hither-to unforeseen things in our daily works and walks. It can be pervasive and still deliver many services without any loss in quality. With technology various devices are being integrated with our already existing environment to make them more comfortable or people friendly. One such scenario has been described here where a unique use case has been presented that can enhance the quality of train travel for travelers through the deeper connectivity and spontaneous integration of a plethora of electronic devices. Here for this example, a wireless sensor network had been dynamically created that would periodically obtain information from various parts of every train coach and contribute for the improvement of the quality of services being provided to the passengers. DPWS, a leading service-based device integration standard specification has been utilized as the device middleware in order to build this integrated application that serves as the foundation in making this use case a reality.

Key words: Traffic wireless sensor networks, DPWS, smart environment, middleware, web services

INTRODUCTION

Smart environments (Augusto *et al.*, 2010; Gabbanini *et al.*, 2012; Kortuem *et al.*, 2010; Chan *et al.*, 2008; Oh *et al.*, 2010; Parra *et al.*, 2009; Rashidi *et al.*, 2011; Zhang *et al.*, 2010) consist of many components such as sensors, actuators and devices that function as inputting, sensing, displaying, controlling, monitoring, management and actuation devices. The brewing trend is that every device (nano, micro to large-scale) is capable of providing computational and communication power. There are displays, monitors and screens for outputting computational results.

Other sophisticated devices such as consumer electronics, cameras, microphones, keyboards, touch screens etc. are also being used today for making our environment more accurate and powerful. The desired outputs can be anything from activating or deactivating a device, sending an emergency event message, showing some content on a public screen, taking a picture of a particular location, or just storing the data in a database for future processing. Data obtained as input can also be combined with preexisting information to produce more relevant results of greater accuracy or make the entire system to work smarter. A variety of tasks get carried out today when neural networks are integrated with such systems and an attempt is made to further enhance the accuracy of these devices or make them more acceptable by the general public. The output can be observed in either the vicinity of the device or somewhere very far off. If the effect or output has to be visible in a far off location, an SMS (short message service), an emergency call, or switch on or off an electronic or electrical device in a remote location may be sent the same way as it was done *in situ*.

These are just some of the ways by which we can use smart technologies today. The transition from information to knowledge goes a long way in making people's lives more cognitive and comfortable. The reach and areas of impacts are being set only by the imagination of the people working in this field. Smart technology can also enhance our experiences in various walks of life. It can be used for creation of smart homes, smart offices, smart hospitals and smarter electricity grids just to name a few.

Various devices were employed here and they can now boast of having made the modern train system smarter. A wireless sensor network (Abielmona *et al*., 2011; Alemdar and Ersoy, 2010; Park *et al*., 2011; Sleman and Moeller, 2008; Xiao *et al*., 2011; Xu *et al*., 2011; Yick *et al*., 2008; Yu *et al*., 2010; Zheng *et al*., 2010) was created which monitored and periodically sent data to the specific devices that were placed in the coaches. Another device which was the central node and also the orchestrator of this entire scenario was installed in an apt place such as the engine of the train. The safety standards of a train were also raised by the proposed system. A number of techniques were created which can increase the feasibility and applicability of the smart technology to further improve, develop and advance the modern train system.

Commuters use public transports such as trains and buses to traverse across various parts of any country on a daily basis. The entire transport system is the backbone of a country's economy. To make a difference to the public transportation system in particular, not only must the entire system be made more efficient and reliable but also make the entire traveling experience more pleasurable and memorable for the passenger/commuter. The system was also made communicative by regularly posting information on the distance of next station along with other available information.

In this paper, usefulness of DPWS as a device middleware (Lee *et al*., 2011; Luo *et al*., 2011; Martin *et al*., 2011; Scherer and Kleinschmidt, 2011) for our scenario has been highlighted. Detailed descriptions about how communication with other devices occurs along with the contributions of features such as WS-Eventing and dynamic device discovery have been provided.

ARCHITECTURE DESCRIPTION

In this section, description about the major components, how they are interacting with one another in order to automate the identified operations, the features and functionalities of each of the modules participating and providing the necessary logic and capability to accomplish the desired actions has been provided.

Device middleware: Middleware is a software that acts like a kind of broker or a mediator. It connects software components or users and their applications. Components such as application servers, web servers and similar tools that support application development and delivery are included. Several modern information technology based on SOAP, XML, Web services (Phaithoonbuathong *et al*., 2010) and Service-oriented Architecture (SOA) are together integrated in a middleware. DPWS (Device Profile for Web Services) (Huang *et al*., 2011; Zeeb *et al*., 2011) is the standard middleware package being leveraged to enable connectivity and integration across domains and devices.

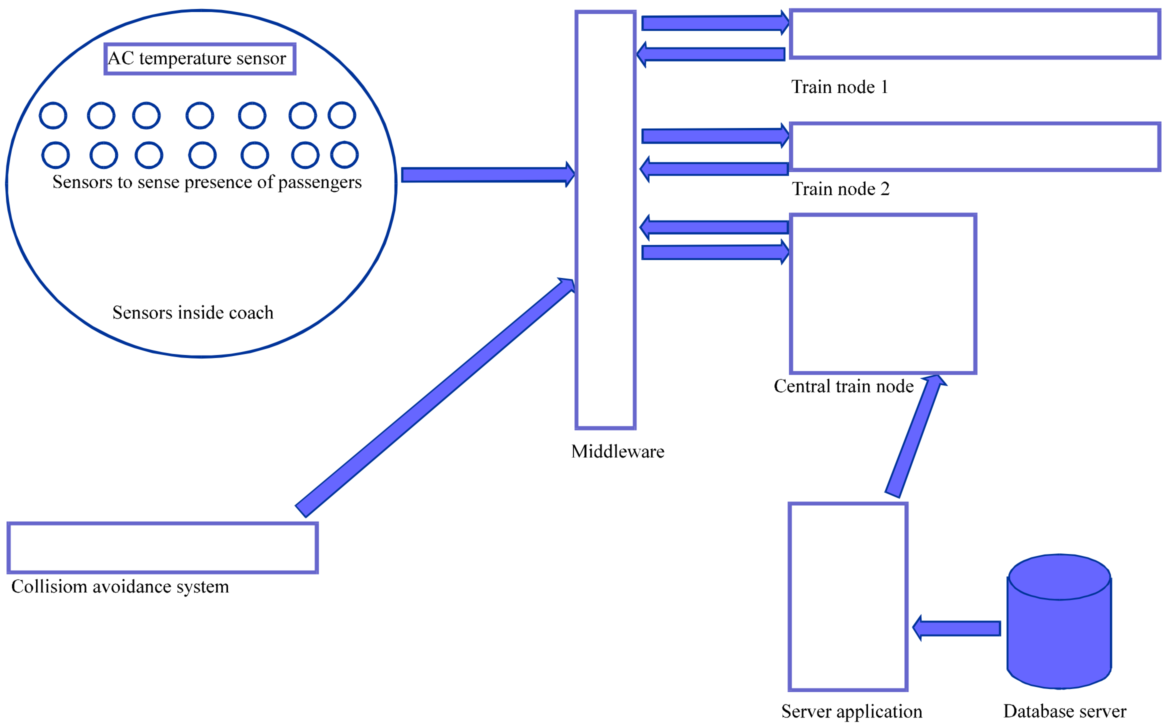

Fig. 1: Data base server communicating with other trains

There are various technologies (Martin *et al*., 2011; Zhang *et al*., 2007) that have been developed in the field of middleware such as OSGi, UPnP and JINI. These technologies can be used for interconnecting devices of various vendors and they also implement SOA. They have some pitfalls of their own, some of which are given here.

OSGi does not support all the available vendors of devices and is entirely dependent on java. A system fully capable of running the Java Virtual Machine (JVM) is needed for OSGi to function. JINI also needs a system capable of running the JVM. Hence there is a reduced scope for use of the technologies such as OSGi and JINI as a suitable middleware. An interpreter such as the JVM cannot be run on resource-constraint devices. In the scenario, use resource-constrained embedded systems were employed. These devices have less power requirements and also have limited processing capabilities. Due to the aforementioned reasons, it was clear that DPWS had to be used, which is designed and developed to cater to such requirements and usages.

Sample scenario: A server with a fully functioning database was set up as shown in Fig. 1. This server is capable of simultaneously communicating with many trains that would be querying it for the information on upcoming stations. An attempt was made to create a server that can handle such requests easily without any degradation in the quality of service, no matter, how many clients were communicating with it (in this case, the trains are the clients.) An RDBMS was used for the purpose of creating and handling the database. The server would retrieve requests from trains and respond by sending the requested information. The database created had tables which contained information regarding the stations and the distance between each station.

The system was made scalable, consistent and reliable by using databases. More tables could be added as required. Security features could be implemented by adding a component that reads the data and encrypts it before transmission to the train. If any further work is done to develop the system, this data could be used or a few more columns could be added to the tables as needed and the services provided can be increased.

There is a central node (a DPWS enabled embedded device with a screen and a few buttons for input) which can be located in the engine or any other suitable location on the train. This device would act like a central repository for the data collected from all the coaches. Data such as the present passenger count and number of empty seats were collected. This device was also connected to the internet which enabled it to connect to a remote server. The remote server provided the information contained in the database.

Each coach would have 2 embedded devices or nodes with a display screen. The first of the devices would show data pertaining to the overall journey such as next station's name, distance of the station. If any station is nearing, it would alert the passengers whose destination is the approaching station that they have to get ready to get down from the train as their journey has come to an end. The other device on each coach showed data collected from the sensors located on the coach. Sensors were placed at apt locations so that necessary data such as the temperature of the AC, the coach number/ID, the number of seats that might be vacant could be collected. This device also received any critical information from the central node on the train pertaining to an obstruction or any emergency due to which the train might be slowing down, or halting.

A small embedded device was developed which was placed in front of the engine. Its sole purpose was to monitor the track and to notify the person controlling the train about a possible chance of an accident. In developing countries such as India, where there are many trains used by a very large number of commuters, safety of the entire train system is very important. The embedded device was included in the developed model keeping this in mind.

The WSN architecture: As mentioned in earlier section, there were certain devices that were installed on every coach. Each coach would have a wireless sensor network of its own. Every node on a coach had a unique ID which could be read by the central node when it first did a probe match and also exchanged the Meta data. (Every device had been identified using a device endpoint address and its IP address. Each coach ID or the coach number was used as the device name.). The central node on a train was the orchestrator for the entire network. It would receive data periodically from the remote server, process it and then send it to the devices on every coach, to be displayed along with the messages related to it. It also retrieved the information from the collision detection device located in front of the engine and showed a message regarding the state of the track. The remote server would transmit the information regarding the distance to next station to the train's central node.

Technologies used

DPWS (Device profile for web services): DPWS was initially published in May 2004. It implements several Web Services specifications while adding features that would make it relatively easier for clients to discover devices offering services. Once a device has been discovered, it is possible for the client toobtain a description of the services hosted on that device. The services can then be used as required. The NET Micro Framework has madeDPWS available to devices that are small and resource constrained. A standard has been created in DPWS that makes it possible to connect devices from different hardware vendors.

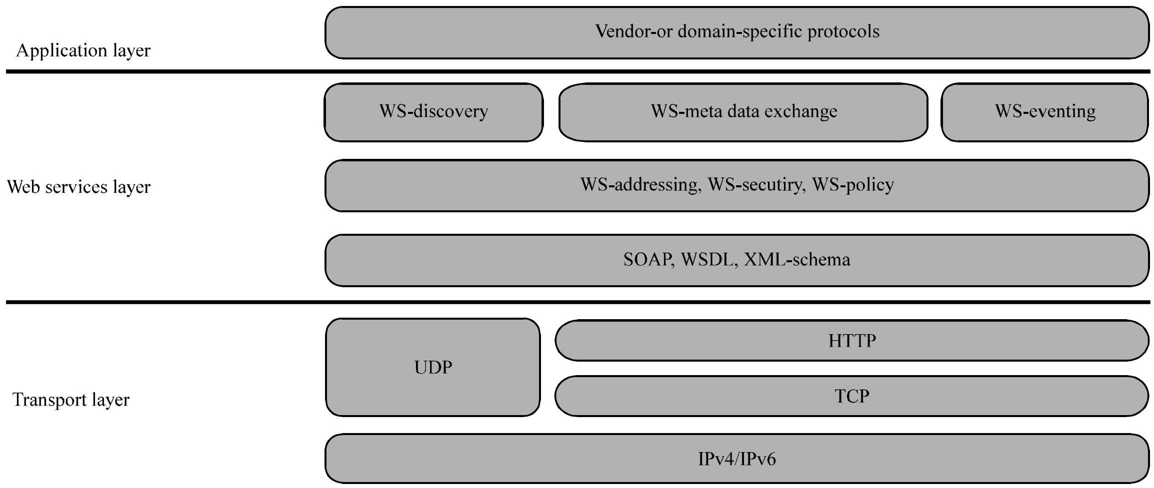

Fig. 2: Major components of DPWS stack

The DPWS stack: The major transport components of DPWS is shown in Fig. 2, are UDP (User Datagram protocol) and TCP/IP(Transmission Control Protocol/Internet Protocol). UDP is used discovery using multicast address and TCP/IP for data exchange. The messages are delivered between a client and server with the HTTP protocol. The HTTP protocolis placed over the TCP/IP. All messages have been formatted in the SOAP (Simple Object Access Protocol), which is based on XML. SOAP is used for a common set of Web Services. Next, there are some important standards from the Web Services technology such as Web Services Addressing, Web Services Security and Web Services Policy. Over these basic components, the standards for discovery, eventing and metadata exchange have been placed.

RDBMS and visual studio: For the purpose of creating and managing databases, MySQL, an elegant, robust and efficient DBMS was used. Microsoft Visual Studio Professional Edition was used for writing and testing of programs in the .Net Micro framework.

The prototype implementation: The first step in the initialization of the system was the device discovery and then obtaining the services offered by it. Probe match was done to discover a device. Every node on the train would first probe for nodes that might be offering services and then obtain their end point addresses. The probe match is a built in feature of DPWS. Probe match could be done as many times as needed. Its results are broadcasted using datagrams.

A wireless sensor network was established. The sensors in this network would read the temperature setting of an AC and return that value to the device. Certain sensors were used to sense the presence of humans in the coach. Pressure sensors were used for this purpose. They were attached to every seat, thereby detecting the presence of a human seated. The output from the sensors was then connected to a DPWS enabled device. The entire data when collected, gives us the availability of free seats. If a seat was sensed by the sensor to be vacant, then it would be displayed on the display screen present in the coach. If needed, the passengers could check it and get permission from the ticketing authority on the train and occupy the seat.

If there were a lot of vacant seats, lights that were not needed would be switched off. This was done by identifying every seat with the light near it. Each sensor on a seat was paired with a device that would be used to switch off the light right on top of that particular seat. The DPWS enabled device would also show messages that were obtained from the central node on the train. If the train had to be stopped, or if any obstruction had been detected, it would receive that information from the central node and display it with a caution symbol. We displayed a red circle that would pop up on the screen with a warning message in case of any obstruction in front of the train.

The other DPWS enabled device would display information regarding the upcoming stations. This information was obtained as a service from the central node. Periodically it would probe the central node for this information and the server would reply with information regarding the stations. While on the journey, if information regarding the areas that the train was going through was available, it might also be displayed.

A database with tables having information on the stations and their distances from the previous location was made. Each table would have data containing station names, starting from a specific starting point of the train. The next column would have the distance of this station from the previous station in kilometers. An application was developed for the server which would first connect to the database and then it would display the entire information on a text field. After a connection with the central node on the train had been made, the information would be sent row by row. The data was sent only upon request because these devices were resource constraint and memory might not be large enough to store the entire data on the device. The information here would be sent to the central node on the train through a TCP/IP connection. TCP/IP was used as it is connected oriented and hence relatively more reliable.

An embedded device was placed in front of the engine to keep track of any obstructions on the track and send a warning if any obstruction was detected. The device had a certain number of proximity sensors which would sense the presence of an obstruction such as another train on the same track, or perhaps a vehicle that might have stopped on the crossing between the road and the track. If any obstruction was detected, message was sent to the central node and the emergency brakes were also activated. This information was also sent to the server located on a remote location. Since the server had a lot of processing power and it also had information on the exact whereabouts of other trains, it could look at the information on other trains. With this information, if there was another locomotive heading towards this train and the person in charge of it has not responded or has failed to see the train, this train can be stopped in time to save the life of the passengers on it. The central node on the train was the orchestration server and it acted like a client when communicating with the remote server.

The node would send the details of the upcoming stations to all the devices that have availed its services. These devices were the ones present on the coaches. Apart from the details of the stations, the node would also get a message periodically from an embedded device that had been installed in front of the engine. This device would send a message, obstruction detected, or no obstruction detected to this central node. Then this data would be sent to the other devices present on the coaches. On the screen, we have displayed images of the engine, for the central node and an image of the interiors of a coach where the device would be installed.

The investigation results: The probe match result has been shown in Fig. 3. After the probe match is performed, the node on the coaches would obtain the endpoint address and the IP address of the central node as shown here. The Database with its tables has been shown Fig. 4. A query was executed to show the entire table and its result is shown on the screen. The application on the server side is shown in Fig. 5.

Fig. 3: Probe match

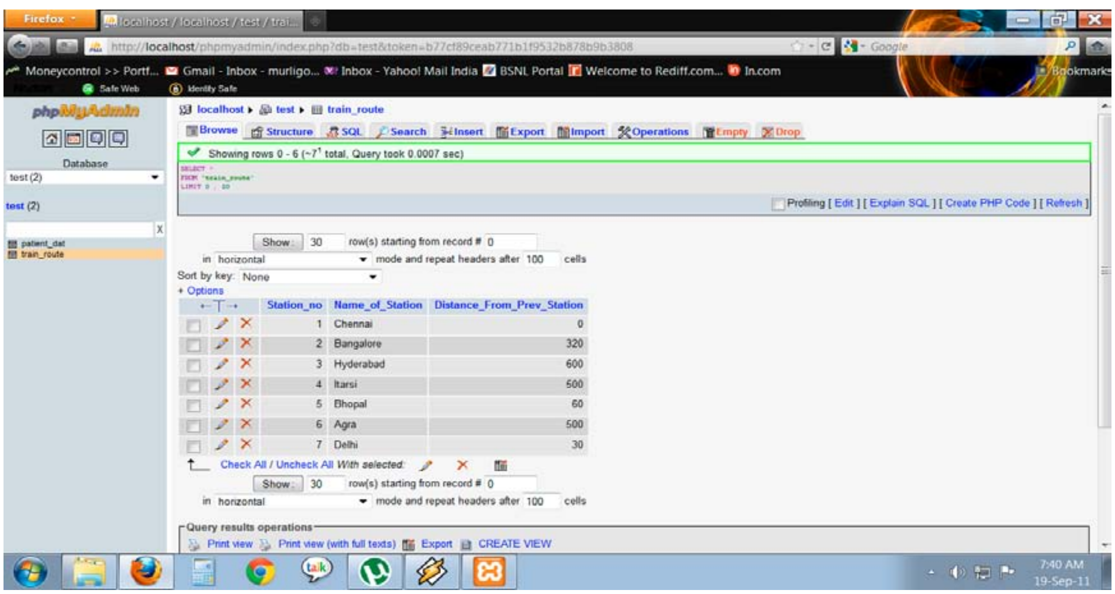

Fig. 4: Table showing the data stored in the database

This data is the same data which is there in the database. After the connection to the train has been established, it would send this data line by line to it. The display screen from the central node and the other device on the coach has been shown in Fig. 6. Two screens have been shown here. One screen is of the central node (the first screen) and the other one is from a node placed on the coach. The other screen on the coach along with the central device has been shown in Fig. 7 and 8. In Fig. 8, the emergency message “obstruction detected” has been shown. This information has also been shown in the screen present on the coach to warn the passengers.

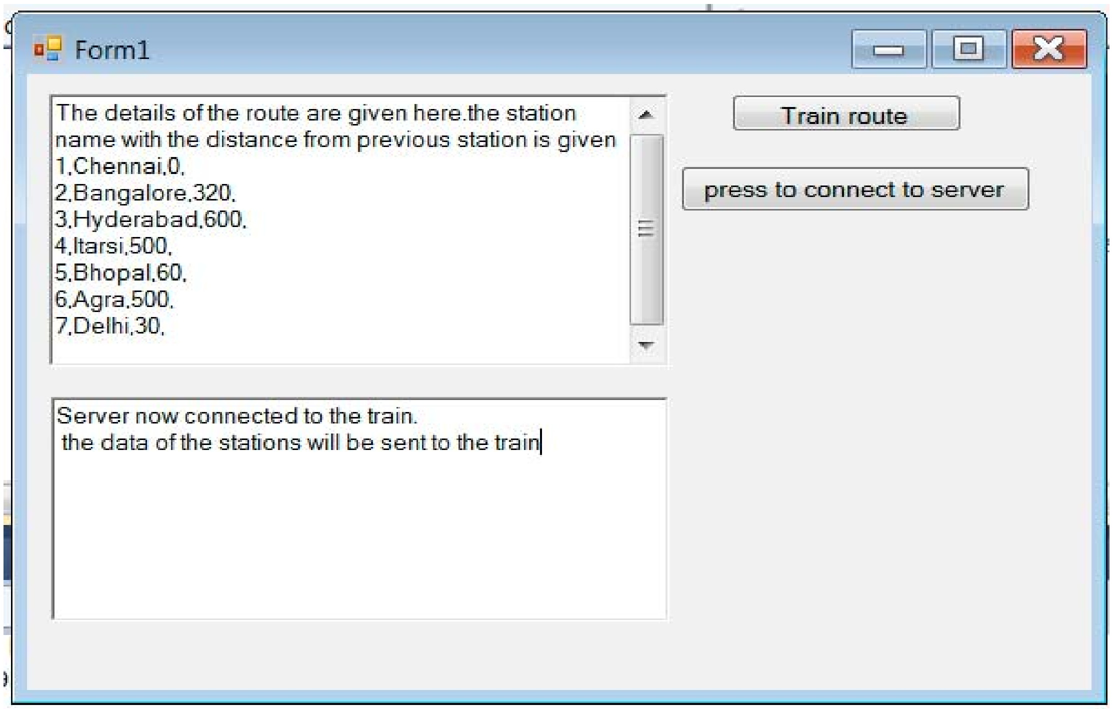

Fig. 5: Central server showing the data obtained from the database

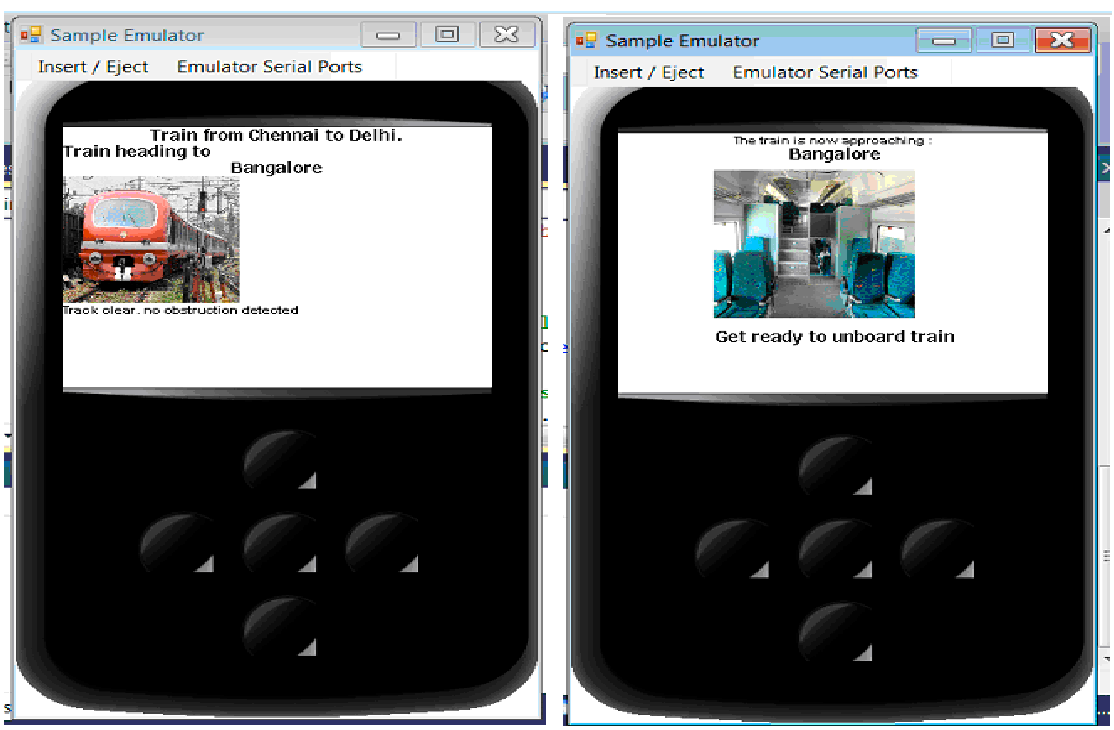

Fig. 6: Central node and a device is placed on a particular coach

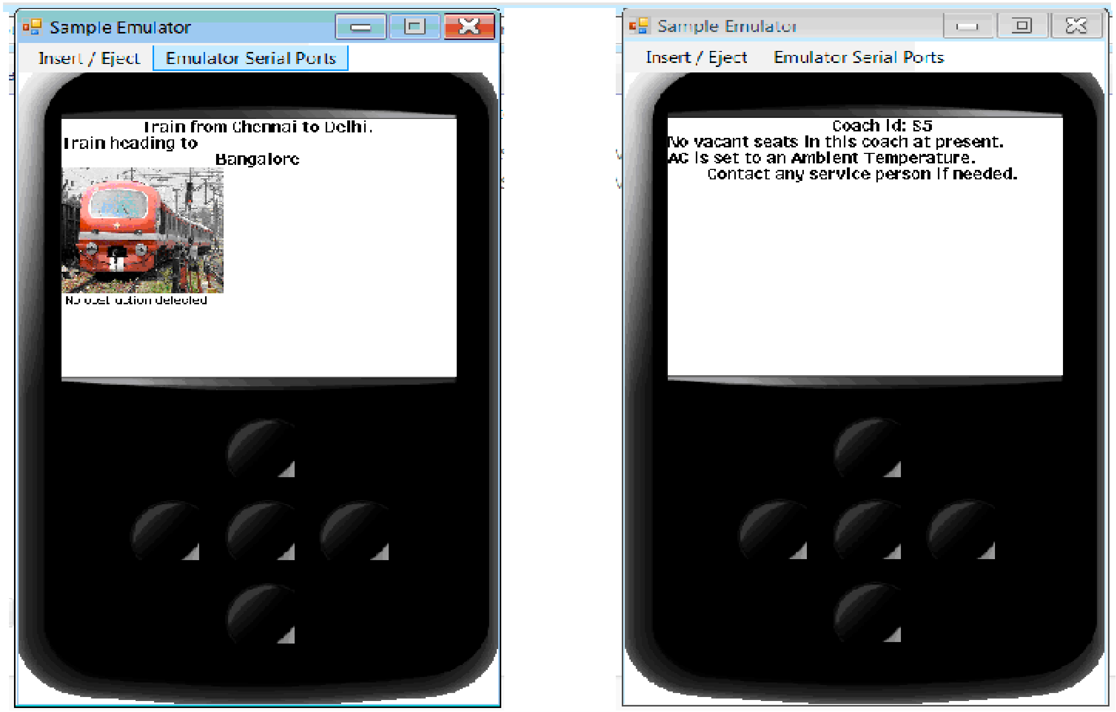

Fig. 7: Central node and the other device placed on the coach display some relevant information

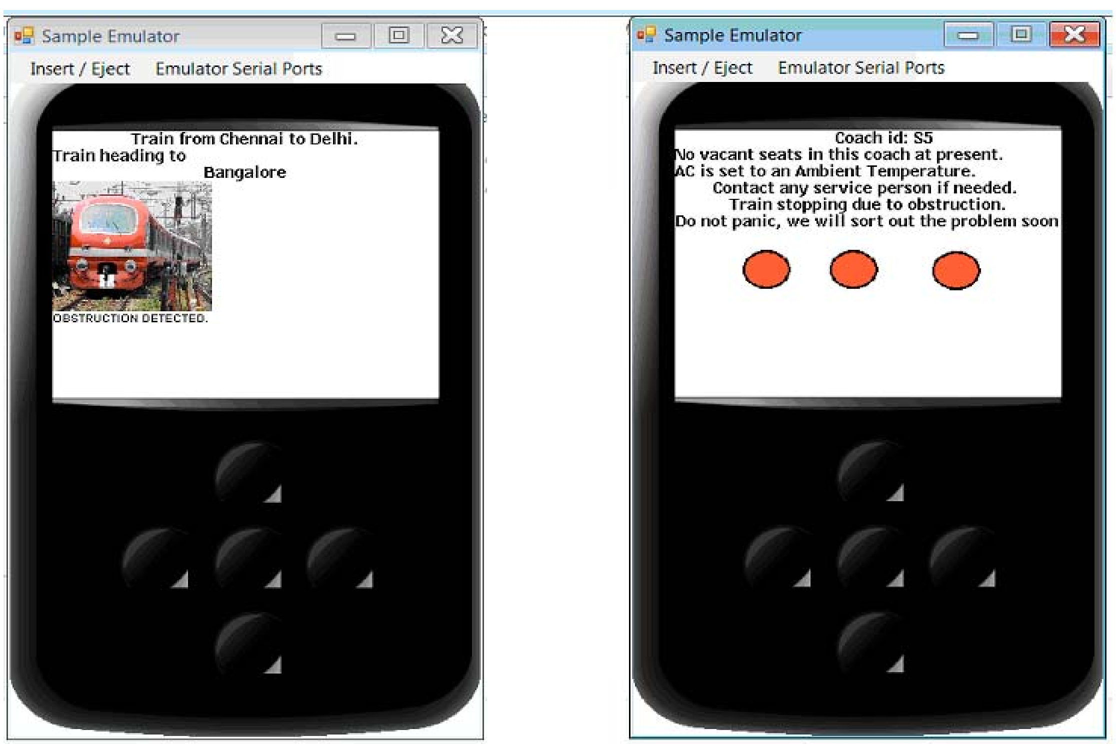

Fig. 8: Central node and the device on coach show an emergency message

CONCLUSION AND FUTURE SCOPE

The modern train system can be improved staggeringly by integrating many novel and nimbler features that would not only make it more secure, but also more gratifying. Today a train journey can last anywhere between a few minutes to several hours. A greater impact would be made if developments are done in some of the crucial aspects such as safety, reliability and extent of comfort. In an attempt to improve the system, we have leveraged a host of smart technologies. We had developed a Wireless Sensor Network (WSN) that was used to make a coach, a place where passengers can obtain any information that might be of use to him. A central node was established which directed the different actions of the entire system. A server was used to send the details of the train's entire route to the central node. An embedded system was designed to check for the proximity of an obstruction and warn the driver of the train and also alert the passengers about it. A DPWS-compliant implementation had been used as the device middleware. All the results obtained were satisfactory.

For future developments, we can try finding out details of a train that might be on the same track. This could be done by querying the remote server regarding any trains that are on the same path. If there is another train, a message from the server alerts its driver. We can integrate the systems of entertainment such as music players that are present on a few luxurious trains to this network. Speakers can be placed on every coach to inform passengers about upcoming stations.

REFERENCES

Abielmona, R., E.M. Petriu and T.E. Whalen, 2010. Distributed intelligent sensor agent system for environment mapping. J. Ambient Intell. Humanized Comput., 1: 95-110.

Alemdar, H. and C. Ersoy, 2010. Wireless sensor networks for healthcare: A survey. Comput. Networks, 54: 2688-2710.

Augusto, J.C., H. Nakashima and H. Ghajan, 2010. Ambient Intelligence and Smart Environments: A State of the Art. In: Handbook of Ambient Intelligence and Smart Environments, Nakashima, H., H. Aghajan and J.C. Augusto (Eds.). Springer, New York, USA., pp: 3-31.

Chan, M., D. Esteve, C. Escriba and E. Campo, 2008. A review of smart homes-present state and future challenges. Comput. Methods Programs Biomed., 91: 55-81.

Gabbanini, F., L. Burzagli and P.L. Emiliani, 2012. An innovative framework to support multimodal interaction with smart environments. Expert Syst. Appl., 39: 2239-2246.

Huang, C.M., H.H. Ku and Y.W. Chen, 2011. Design and implementation of a web 2.0 service platform for dpws-based home-appliances in the cloud environment. Proceedings of the IEEE Workshops of International Conference on Advanced Information Networking and Applications, March 22-25, 2011, Biopolis, Singapore, pp: 159-163.

Kortuem, G., F. Kawsar, D. Fitton and V. Sundramoorthy, 2010. Smart objects as building blocks for the internet of things. IEEE Internet Comput., 14: 44-51.

Lee, D., T. Kim, S. Jeong and S. Kang, 2011. A three-tier middleware architecture supporting bidirectional location tracking of numerous mobile nodes under legacy WSN environment. J. Syst. Archit., 57: 735-748.

Luo, J., L. Chen and L. Chen, 2011. A tactic agent middleware in wireless sensor networks based on associative monitoring scheme. Int. J. Digital Content Technol. Appl., 5: 79-86.

Martin, S., G. Diaz, I. Plaza, E. Ruiz, M. Castro and J. Peire, 2011. State of the art of frameworks and middleware for facilitating mobile and ubiquitous learning development. J. Syst. Software, 84: 1883-1891.

Oh, Y., J. Han and W. Woo, 2010. A context management architecture for large-scale smart environments. IEEE Commun. Mag., 48: 118-126.

Park, H., J. Lee, S. Park, S. Oh and S. Kim, 2011. Multicast protocol for real-time data dissemination in wireless sensor networks. IEEE Commun. Lett., 15: 1291-1293.

Parra, J., M.A. Hossain, A. Uribarren, E. Jacob and A.E. Saddik, 2009. Flexible smart home architecture using device profile for web services: A peer-to-peer approach. Int. J. Smart Home, 3: 39-55.

Phaithoonbuathong, P., R. Monfared, T. Kirkham, R. Harrison and A. West, 2010. Web services-based automation for the control and monitoring of production systems. J. Comput. Integr. Manuf., 23: 126-145.

Rashidi, P., D.J. Cook, L.B. Holder and M. Schmitter-Edgecombe, 2011. Discovering activities to recognize and track in a smart environment. IEEE Trans. Knowl. Data Eng., 23: 527-539.

Scherer, R. and J.H. Kleinschmidt, 2011. A middleware architecture for wireless sensor networks using secure web services. IEEE Latin Am. Trans., 9: 815-820.

Sleman, A. and R. Moeller, 2008. Integration of wireless sensor network services into other home and industrial networks; using Device Profile for Web Services (DPWS). Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, April 7-11, 2008, Damascus, Syria, pp: 1-5.

Xiao, H., Y. Gong, H. Ogai, J. Zhang, X. Zou, T. Otawa and T. Tsuji, 2011. A data collection system in wireless network integrated WSN and ZIGBEE for bridge health diagnosis. Proceedings of the SICE Annual Conference, September 13-18, 2011, Tokyo, Japan, pp: 2024-2028.

Xu, X., H. Shen and Y. Ling, 2011. A novel method for indoor wireless sensor network localization. J. Comput. Inform. Syst., 7: 4522-4530.

Yick, J., B. Mukherjee and D. Ghosal, 2008. Wireless sensor network survey. J. Comput. Network, 52: 2292-2330.

Yu, H., Z. Shen, C. Miao, C. Leung and D. Niyato, 2010. A survey of trust and reputation management systems in wireless communications. Proc. IEEE, 98: 1755-1772.

Zeeb, E., G. Moritz, D. Timmermann and F. Golatowski, 2010. Ws4d: Toolkits for networked embedded systems based on the devices profile for web services. Proceedings of the 39th International Conference on Parallel Processing Workshops, September 13-16, 2010, San Diego, CA., USA., pp: 1-8.

Zhang, Q., X.L. Yang, Y.M. Zhou, L.R. Wang and X.S. Guo, 2007. A wireless solution for greenhouse monitoring and control system based on ZigBee technology. J. Zhejiang Univ. Sci. A, 8: 1584-1587.

Zhang, S., S. McClean, B. Scotney, X. Hong, C. Nugent and M. Mulvenna, 2010. An intervention mechanism for assistive living in smart homes. J. Ambient Intell. Smart Environ., 2: 233-252.

Zheng, G., S. Liu and X. Qi, 2010. Survey on topology of wireless sensor networks based on small world network model. Control Decis., 25: 1761-1768.