SAS 2007 - IEEE Sensors and Applications Symposium
San Diego, California, USA, 6-8 February 2007.

An Implementation of Smart Transducer Web Services for IEEE 1451-based Sensor Systems

Eugene Song, Kang Lee
National Institute of Standards and Technology
100 Bureau Drive, MS# 8220
Gaithersburg, Maryland USA 20899-8220
Phone: (301) 975-6542, (301) 975-6604
E-Mail: ysong@nist.gov, kang.lee@nist.gov

Abstract *- This paper introduces an implementation of a proposed Smart Transducer Web Services developed at the National Institute of Standard and Technology (NIST) based on IEEE P1451.0 standard. This implementation system consists of a service consumer, service provider, and wireless sensor node. A service consumer and provider can communicate with each other using Smart Transducer Web Services through SOAP (Simple Object Application Protocol) messages. Service consumers in this system can discover transducers, access transducer data, and retrieve transducer TEDS (Transducer Electronic Data Sheets) using Smart Transducer Web Services. Both service provider and consumer are generated from the developed WSDL (Web Service Description Language) file. The service provider and wireless network node is implemented based on the IEEE P1451.0 and P1451.5 standards using the 802.11 wireless communication protocol. The work described in this paper provides a good foundation for Smart Transducer Web Services standardization and provides a standard way for IEEE 1451 applications to easily facilitate interoperability with other sensor applications by means of the developed WSDL file.*

Keywords: IEEE 1451.0, IEEE 1451.5, NCAP, Service-oriented Architecture, Smart Transducer Web Services, SOAP, TEDS, TIM, Web Services, WSDL, XML

I. INTRODUCTION

The evolution of sensor networks has been fueled by technological innovations from the sensing, computer, and communications industries. The growth of these networks has accelerated to meet demands for the use of sensors in a wide range of applications and for enterprise-wide exchange of information [1]. The increased availability of communications and networking systems (both wired and wireless) and the demand for ubiquitous communications is likely to bring about a crossover of price and functionality between sensor networking technology and communications technology. In addition, low-power wireless communications technology will make it possible to overcome the need to wire in key applications. The growing dissemination and use of World Wide Web browsers will also have a significant impact on sensor networking.

Service-oriented architecture (SOA) is an evolution of distributed computing based on the request/response design paradigm for synchronous and asynchronous applications [2]. Services are software components that have published interfaces; these interfaces are platform, language, and operating system independent. The most important aspect of a service is the service description using Web Services Description Language (WSDL) that describes the types, messages, and operations of the web service, and is the contract to which the web service guarantees it will conform [3]. SOA provides ways for standard-based, platform, language, and operating system independent interoperability between software applications. Applying SOA to sensors and sensor networks is a very important step forward to presenting the sensors as reusable resources, which can be discovered, accessed, and controlled via the Internet.

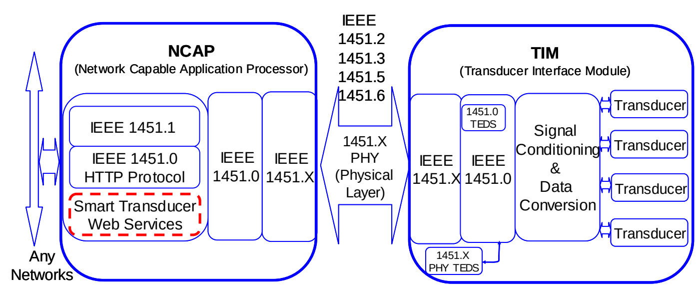

Figure 1 IEEE 1451 family standards

The IEEE 1451 family of standards define a set of common communication interfaces for connecting transducers (sensors or actuators) to microprocessor-based systems, instruments, and field networks in a network-independent environment. They provide a set of protocols for wired and wireless distributed applications [4]. Figure 1 shows the family of the IEEE 1451 standards. The proposed IEEE 1451.0 standard, designated as IEEE P1451.0 [5], defines a set of common commands, electronic data sheet formats, and communication protocols for the family of IEEE 1451 standards. One of the goals of IEEE P1451.0 is to help achieve data-level interoperability for the 1451 family. There are three possible ways to access transducers via a network, 1) IEEE 1451.1 standard, 2) IEEE P1451.0 HTTP protocol, and 3) proposed Smart Transducer Web Services. This paper mainly focuses on the IEEE P1451.0-based Smart Transducer Web Services.

II. RELATED WORK

The Sensor Web is a new class of geographic information system (GIS) developed at NASA's Jet Propulsion Laboratory consisting of a sensor network for environmental monitoring and control [6]. Some sensors are already on the Web and able to return their location information as well as observations and measurements. The missing element for Sensor Web is a universal standard framework for describing and tasking sensors in XML (eXtensible Markup Language) [7]. The Open Geospatial Consortium (OGC) Sensor Web Enablement (SWE) initiative focuses on developing standards to enable the discovery and exchange of sensor observations as well as the tasking of sensor systems. OGC members have developed and tested the following candidate specifications: Observations & Measurements (O&M), Sensor Model Language (SensorML), Transducer Model Language (TML), Sensor Observations Service (SOS), Sensor Planning Service (SPS), Sensor Alert Service (SAS), and Web Notification Services (WNS) [8]. The IEEE 1451 standards deal with sensor information from physical sensors to the network level, while OGC-SWE takes the sensor information and brings it into applications, in particular, via the web. There is a great opportunity to apply these two sets of standards together to ultimately achieve the ease of use of sensors and ability to transfer sensor information from sensors to application in a seamless way using consensus-based standard [9]. SensorNet is a framework to support plug-and-play sensors of various types, archival storage of sensor data, standards-based publication of sensor data, and sensor control services, and to tie together sensor data from all over the country to create a real-time detection and alert system for various threats, whether they are chemical, radiological, biological, nuclear, or explosive (CBRNE) [10,11]. SensorNet's purpose is to influence and test IEEE 1451 and OGC-SWE specifications and other relevant standards to provide a harmonization solution of sensor standards and a set of best practices for a comprehensive, nationwide system for real-time detection, identification, and assessment of CBRNE hazards. This work has indicated that SOA is the best way to seamlessly integrate IEEE 1451 with OGC-SWE standards and other sensor applications. A web-service based on IEEE 1451.1 standard is described in the reference [12, 13]. The IEEE P1451.0 standard provides a common set of commands, electronic data sheet formats, and communication protocols for the family of IEEE 1451 standards. Smart Transducer Web Services based on the IEEE P1451.0 standard are a good solution for IEEE 1451 applications to seamlessly integrate and to do interoperability with sensor alert, OGC-SWE or any other sensor applications.

III. IMPLEMENTATION SYSTEM Of SMART TRANSDUCER WEB SERVICES

A. Architecture of Smart Transducer Web Services for IEEE 1451

The Web Services Interoperability (WS-I) Organization is an open, industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages. WS-I provides supporting implementation guidance and testing resources for each of the profiles it develops [14].

Figure 2 shows the architecture of a smart transducer service for the IEEE 1451 standard in order to do standard-based interoperability. An IEEE 1451 NCAP (Network Capable Application Processor), on which the web server runs, can be used as a transducer service provider. Each NCAP provides a set of 1451 Smart Transducer Web Services. Service consumers, such as a sensor alert system, OGC-SWE, or any other sensor application, can find the Smart Transducer Web Services and then invoke these Smart Transducer Web Services through SOAP/XML messages. A service consumer and service provider can communicate with each other by means of WSDL through SOAP/XML messages. The service consumer sends a request for sensor data to the service provider. When the service provider receives a request from the service consumer, it invokes smart transducer services based on the request and communicates with the TIM (transducer interface module) for sensor data through IEEE 1451.x communication module. The TIM then returns the sensor data (result) to the service provider (NCAP). Finally, the service provider sends sensor data back to the service consumer using a SOAP/XML message.

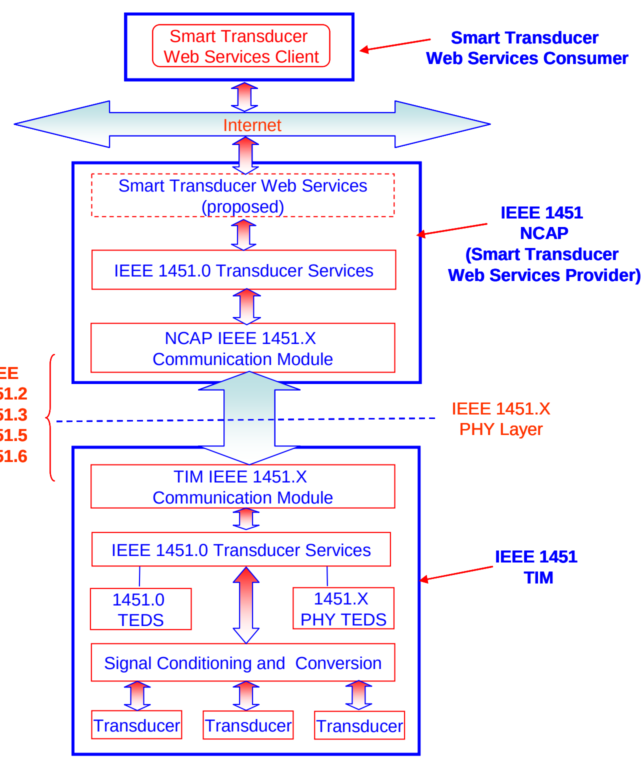

Figure 2. Architecture of Smart Transducer Web Services for IEEE 1451

B. Smart Transducer Web Services Description in WSDL

The transducer service interface of IEEE P1451.0 is used by measurement and control applications to access the IEEE P1451.0 system. This interface contains the operations to read and write transducer channels, read and write TEDS, and send configuration, control, and operation commands to the TIMs. The transducer services interface of IEEE P1451.0 contains six interfaces: TransducerAccess, TransducerManager, TimDiscovery, TedsManager, CommManager and AppCallback.

- TimDiscovery: Discover available IEEE 1451.x communications modules, TIMs, and TransducerChannels.
- TransducerAccess: Access sensor and actuator TransducerChannels.
- TransducerManager: Control over TIM access.
- TedsManager: Read and write TEDS. This class also manages the NCAP-side TEDS cache information.
- CommManager: Handle access to the communication module on the local device.
- AppCallback: Applications that have advanced features need to implement this interface.

WSDL is an XML format for describing Web services as a set of endpoints operating on messages. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. WSDL specifies the location of the service and the operations (or methods) the service exposes. The WSDL specification is divided into six major elements: definitions, types, message, portType, binding, and service. Smart Transducer Web Services described in WSDL include service definition, types of IEEE P1451.0 standard, message, portType (a set of operations), service binding with supported protocol, and service port. The types described in XML schema include data types and TEDS of IEEE P1451.0 and data types defined in the service requests and response messages. The portType is defined based on the IEEE P1451.0 transducer services. The detailed description of Smart Transducer Web Services is included in the reference [16].

Figure 3 shows the interfaces of a successfully deployed WSDL file of the Smart Transducer Web Services. The Smart Transducer Web Services interfaces include TimDiscovery, TransducerDiscovery, ReadTransducerData, and ReadTimMetaIDTedsData. The Smart Transducer Web Services are described in WSDL to facilitate interoperability with sensor applications.

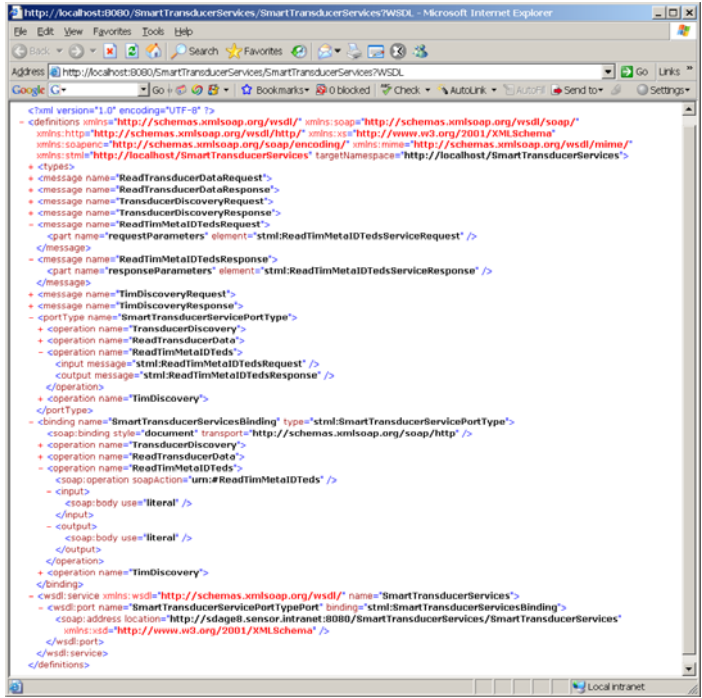

Figure 3. Interfaces of a successfully deployed WSDL file

C. Implementation System of Smart Transducer Web Services

Figure 4 shows an implementation system of Smart Transducer Web Services. This system consists of a service consumer, service provider (a NCAP), and wireless sensor node (a TIM). The NCAP, on which the web server resides, is used as a service provider. The NCAP wirelessly communicates with the TIM through the IEEE P1451.0 and P1451.5 protocols using the client-server and publisher-subscriber communication model. The client-server and publisher-subscriber communications among these nodes were implemented using TCP/IP (Transmission Control Protocol/Internet protocol) and TCP/UDP (User Datagram Protocol), respectively. The IEEE 1451.2-based sensors are connected to the TIM through a serial port. The service consumer finds and locates the Smart Transducer Web Services and then invokes these web services on the NCAP using SOAP/XML messages. In essence, the service consumer and service provider communicate with each other using SOAP/XML messages.

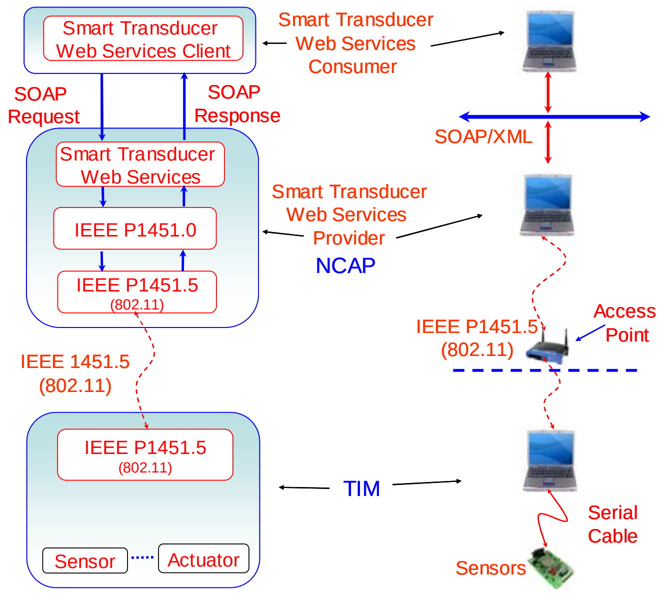

Figure 4. An implementation system of Smart Transducer Web Services

The service request–response process of Smart Transducer Web Services can be described as follows: 1) a service consumer sends a request to a service provider through a SOAP/XML message; 2) the service provider processes the request and then calls the corresponding IEEE P1451.0 transducer service of the NCAP; 3) the NCAP calls the IEEE P1451.5 communication module on the NCAP; 4) the IEEE P1451.5 communication module on the NCAP communicates with the IEEE P1451.5 communication module on the TIM through the IEEE P1451.5-802.11 communication module and get the result from the TIM [15]; 5) the IEEE P1451.0 web service on the NCAP gets the result from the IEEE P1451.5 communication module on the NCAP; 6) the service provider gets the result from the IEEE P1451.0 and forwards it to the service consumer through a SOAP/XML message.

C.1. Smart Transducer Web Services Provider

The Smart Transducer Web Services provider is generated from the developed WSDL file using a web service development tool. Creating a web service provider from the WSDL file involves the following steps: 1) specify a name for the web service provider; 2) specify the location of the WSDL file; 3) the tool automatically generates a web service; 4) implement the web service by adding source code, for example, to implement TIM discovery or read TEDS function; 5) test web services; and 6) deploy the web services. Figure 5 shows the interface of the Smart Transducer Web Services provider. The interface includes four web services: TimDiscovery, TransducerDiscovery, ReadTransducerData, and ReadTimMetaIDTeds. The implementation of the Smart Transducer Web Services provider based on the IEEE P1451.0 and P1451.5 standards is described in the reference [17].

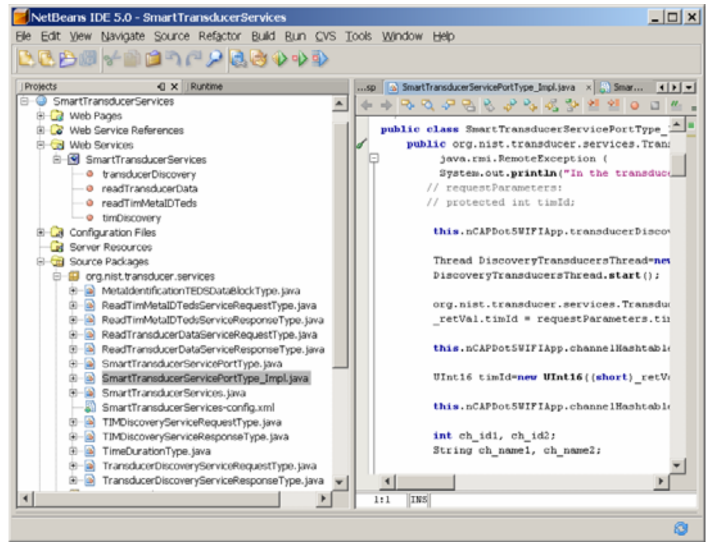

Figure 5. Interface of Smart Transducer Web Services provider

C.2. Smart Transducer Web Services Consumer

The Smart Transducer Web Services consumer is generated from the developed WSDL file using a web service development tool. Creating a web service consumer involves the following steps: 1) specify a name for the web services consumer; 2) specify the location where the WSDL file was deployed by the provider; 3) the tool automatically generates a web services client; 4) develop a specific web services client by adding source code such as a JSP (Java Server Page), Servlet, or Java application, to invoke or call the web services; 5) test web service client; and 6) deploy the web services client. Figure 6 shows the interface of Smart Transducer Web Services consumer generated using WSDL.

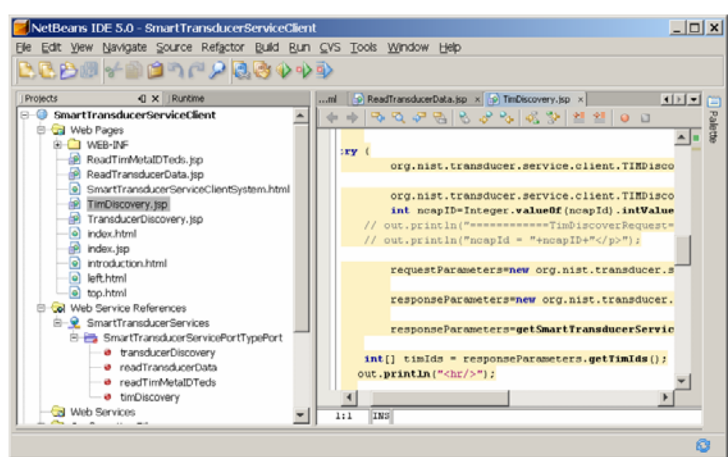

Figure 6. Interface of Smart Transducer Web Services consumer

IV. CASE STUDIES OF SMART TRANSDUCER WEB SREVICES

The IEEE 1451.2 standard defines a transducer interface for connecting sensors and actuators to microprocessors, instruments, and networks [18]. Accessing IEEE 1451.2 sensor data and TEDS data are taken as examples to test the Smart Transducer Web Services system based on the IEEE P1451.0 and P1451.5 standards. When a TIM starts up, it automatically announces itself to the NCAP, and the NCAP discovers it. The following steps take place: 1) the communication module of the TIM registers to the TIM and the communication module of NCAP registers to the NCAP, respectively; 2) the TIM does a TIM announcement to the NCAP; 3) the NCAP does a TIM discovery. The TIM announcement operation is described in detail in the reference [16].

A. Case Study of TIMDiscovery Service

The TIM discovery service discovers all the TIMs announced to the NCAP, and gets all the timIds of the TIMs. Figure 7 shows the user interface of TIMDiscovery service. This result illustrates that there is one NCAP with ncapId (1). This NCAP has one TIM with a timId (165).

Figure 7. User interface of TIM discovery service

B. Case Study of TransducerDiscovery Service

After discovering the TIMs, the TransducerDiscovery service can be invoked. The TransducerDiscovery service gets all channelIds and channelNames of the specified TIM. Figure 8 shows the user interface of the TransducerDiscovery service. This result illustrates that the TIM with the timId (165) has two transducer channels. The first transducer channel is an IEEE 1451.2-based Hall-effect sensor with the channelId (1). The second transducer channel is an IEEE 1451.2-based illumination sensor with the channelId (2).

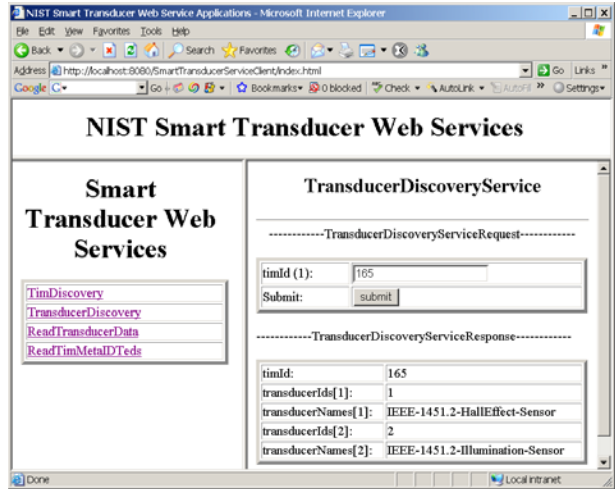

Figure 8. User interface of transducer discovery service

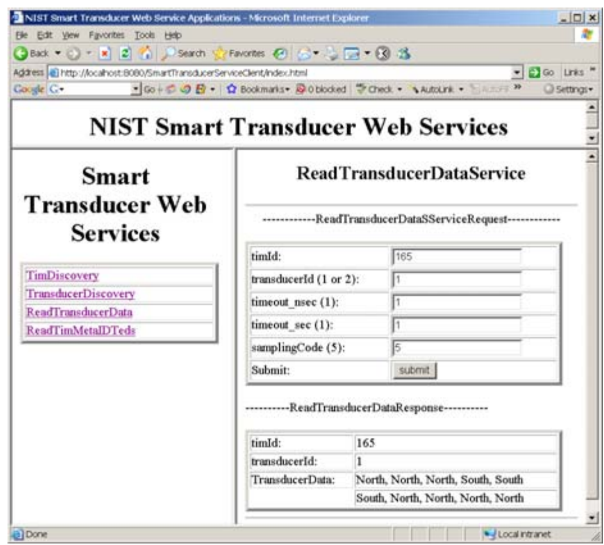

Figure 9. Channel #1 Hall-effect sensor data reading

C. Case Study of ReadTransducerData Service

Figure 9 shows the user interface of the ReadTransducerData service. The display shows the Hall-effect sensor data with the transducerId (1). The Hall-effect sensor data illustrate the results of sensing magnetic pole changes.

Figure 10 shows the user interface of the ReadTransducerData service. The display shows the illumination sensor data with a transducerId (2). The illumination sensor data illustrates relative light intensity readings.

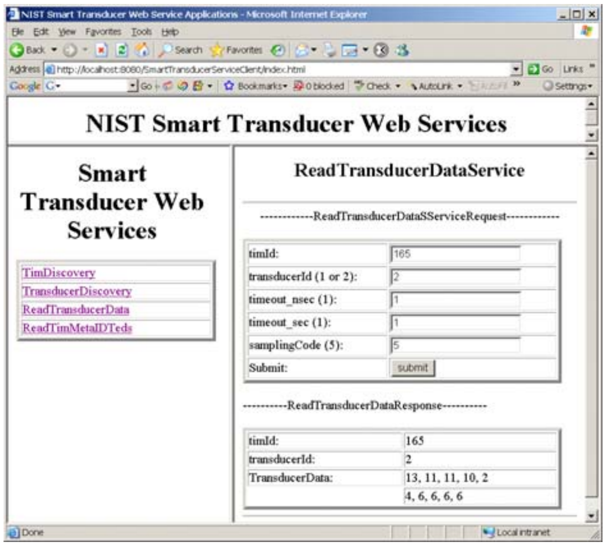

Figure 10. Channel #2 illumination sensor data reading

V. CONCLUSION

This paper introduces an implementation system of the proposed Smart Transducer Web Services developed at NIST. This system consists of a service provider, service consumer, and wireless sensor node. Both service provider and consumer are generated from the developed WSDL file. The service consumer and provider communicate with each other using the Smart Transducer Web Services through SOAP messages. The service provider, a wireless network node, is implemented following the IEEE P1451.0 and P1451.5 standards and using the IEEE 802.11 wireless communication modules to access sensor data, and then forwards the data to the service consumer through SOAP messages. The service consumer discovers sensors and accesses sensor data and their TEDS data using the Smart Transducer Web Services. The work described in this paper provides a standard-based approach to facilitate interoperability among IEEE 1451-based and other sensor applications by means of the developed WSDL.

VI. REFERENCE

1. Jay Warrior. Smart Sensor Networks of the Future: http://archives.sensorsmag.com/articles/0397/net_mar/main.shtml
2. http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html
3. http://www.onjava.com/pub/a/onjava/2005/01/26/soa-intro.html
4. Lee, K., “IEEE 1451: A Standard in Support of Smart Transducer Networking”, Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference, Baltimore, MD, May 1-4, 2000, Vol. 2, pp. 525-528.
5. IEEE P1451.0D6.003, Draft Standard for a Smart Transducer Interface for Sensors and Actuators– Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016.
6. http://en.wikipedia.org/wiki/Sensor_Web
7. Mark Reichardt. The Sensor Web's Point of Beginning. http://www.geospatial-online.com/geospatialsolutions/article/articleDetail.jsp?id=52681
8. Mike Botts, George Percivall, Carl Reed, and John Dividson. OGC Sensor Web Enablement: Overview and High level Architecture – an OGC white paper.
9. Kang B. Lee, Mark E. Reichardt, Open standards for homeland security sensor networks – sensor interconnection through web access. IEEE Instrumentation and measurement magazine, Dec. 2005, pp 20-26.
10. http://www.sensornet.gov/sn_overview.html
11. Bryan L. Gorman, Mallikarjun Shankar, and Cyrus M. Smith, Advancing sensor web interoperability. http://www.sensorsmag.com/sensors/article/articleDetail.jsp?id=185897
12. Eamil F. Sadok, Ramiro Liscano, A web-service framework fro 1451 sensor networks. IMTC 2006 –Instrumentation and measurement technology conference, Ottawa, Canada, 17-19 May 2005
13. Vitor Viegas, J.M.Dias Pereira, P. Silva Firao, IEEE 1451.1 standard and XML web srevices: a powerful combination to build distributed measurement and control systems. IMTC 2006 – Instrumentation and measurement technology conference, Sorrento, Italy 24-27, 2006
14. Web service Interoperability: http://www.ws-i.org/about/Default.aspx
15. IEEE P1451.5D10.1, Draft Standard for a Smart Transducer Interface for Sensors and Actuators– Wireless Communication and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016.
16. Eugene Song, Kang Lee, Smart Transducer Web Services based on IEEE P1451.0 Standard IMTC 2007 - Instrumentation and Measurement Technology Conference, WARSAW, POLAND, MAY 1-3, 2007. (to be submitted)
17. Eugene Song, Kang Lee, An implementation of proposed IEEE 1451.0 and 1451.5 standards, IEEE Sensors and Applications Symposium, Houston, Texas, USA, February 7-9, 2006, pp. 72-77.
18. IEEE STD 1451.2-1997, Standard for a Smart Transducer Interface for Sensors and Actuators–Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH94566, Sept. 25, 1998.