Computer Standards & Interfaces 38 (2015) 78–83

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Service-oriented implementation model for smart transducers network

Nikola Bezanic *, Ivan Popovic

School of Electrical Engineering, University of Belgrade, Bul. Kralja Aleksandra 73, 11120 Belgrade, Serbia

ARTICLE INFO

Article history:
Received 20 September 2013
Received in revised form 9 October 2014
Accepted 15 October 2014
Available online 24 October 2014

Keywords:
Service Oriented Architecture
Smart Transducer Web Services
Virtual Transducer
IEEE 1451 Smart Transducer
Web Service

ABSTRACT

This paper describes a new implementation model for the service-oriented smart transducers network based on the IEEE 1451 Web Services. The model enables simple service dislocation and addition of new functionalities in Service Oriented Architecture (SOA) network. Presented architectural organization supports new service-oriented network entities in addition to standard IEEE 1451 smart transducers. The entities such as particular transducer services and functionalities, processing applications and algorithms, and set of I/O devices are supported in the form of service providers managed by a central server. The entities are modeled as virtual transducers incorporated in the service-oriented network, and analysis on how this architectural change affects the smart transducer design constraints is given. The case study of a smart transducer network design in form of automated configuration and data exchange between ARM-based smart transducer interface module, central server and dislocated virtual transducer is presented.

1. Introduction

Transducers play an important role in applications in science and industry as they can be used for monitoring and controlling physical phenomena. In general, transducer may be either a sensor or an actuator; where sensor is used to transform a parameter being measured to an electrical signal, while actuator takes a physical action based on a received electrical signal. Reading sensors (or writing to actuators) is made easier to some extent with introduction of digital interfaces in a way that it is possible to establish an efficient system communication using standard wireless transceivers or wired serial interfaces. Device that integrates sensors/actuators, signal conditioning and amplification circuits, AD/DA converters, a processing unit,and a communication interface is known as smart transducer [1]. It is considered to be a small, compact hardware/software unit where processing application can be implemented using a microcontroller with associated software. Also, computer networking has been used in transducer systems for more than ten years, resulting in a common platform for flexible installation and maintenance of smart transducers. Contemporary microcontrollers with network support are able to integrate the majority of smart transducer building blocks, as will be explained later in this work.

In order to make the smart transducer manufacturing more consistent, regulatory bodies made a joint effort with industry to propose the IEEE 1451 family of standards [2]. Specification marked as IEEE 1451.0 describes the common features of smart transducers: common functions, communication protocols, and Transducer Electronic Data Sheets (TEDS). Smart transducer networking is also supported by the standard, but in particular, proposed interfaces do not address the network interoperability problem. An example of support for the IEEE 1451 devices is presented in [3], as a hardware/software co-design platform based on FPGA. However, presented solution does not provide a support for the efficient network data exchange. Interoperability can be improved using the Service Oriented Architecture (SOA) inside the smart transducers network. SOA can be implemented by means of Web service technology where each service is wrapped around particular IEEE 1451.0 smart transducer function, as described in [4]. This way, client application communicates with smart transducer by querying its services over the network. Message format specification is written in XML using Web Service Description Language (WSDL). Using Web services in real time scenarios such as environmental monitoring or intelligent transportation systems (ITS) represents an emerging trend [5,6]. This way, application component can publish its operations and be easilyintegrated in a distributed environment. Applying Web services to smart transducers represents a new paradigm known as Smart Transducer Web Services (STWS) [7].

As a part of the Open Geospatial Consortium-Sensor Web Enablement (OGC-SWE) initiative, set of standard interfaces are defined for accessing sensor data in a network. Study [8] proposes architectural design of a real-time environmental sensor network based on OGC-SWE standards. In this case data are easily accessible at the network level, but problem of getting data from physical devices and their efficient injection into the network is not solved.

STWS services can be used for efficient sensor data access and data transfer towards OGC-SWE applications as shown in prototype [4]. The prototype does not solve a problem of embedded Web services implementation. Embedded smart transducer that integrates Web services

* Corresponding author. Tel.: +38 1641845140.
E-mail addresses: nbezanic@gmail.com (N. Bezanic), popovici@el.etf.rs (I. Popovic).

http://dx.doi.org/10.1016/j.csi.2014.10.004
0920-5489/

is able to communicate directly with its service client over the network, without intermediary implementation.

Ice detection system for road monitoring is described in [9]. Innovative ice sensor and salinity sensor are designed, and ice forecasting methods are presented together with experimental results. Interoperability between sensor devices and forecasting algorithms is not particularly taken into account.

This paper proposes new architectural changes for improving interoperability of the entities in smart transducers network and simplifying its implementation. Presented concept of the IEEE 1451 services implementation and data exchange in smart transducer network enables smart transducer design on small-memory embedded platform with limited processing resources through migration of services from embedded to general purpose platform. Virtual transducers provide a flexible platform for implementationof the dislocated processing application, including both IEEE 1451 and application specific functionalities. This way, it is possible to reduce hardware requirements of embedded system and optimize its power consumption.

2. System Overview

2.1. Virtual smart transducer

In this paper, we build on the STWS interoperability potential [7], which offers a uniform way of accessing data as interface is once established by WSDL. We go one step further and introduce a new entity behind the already defined Web services. To achieve symmetry between transducers and their applications, we introduce a concept of virtual transducers, similar as in [10]. Hence, virtual smart transducer is a data processing application equipped with STWS interface, but rather acts as a STWS service producer instead of a common consumer. This setup requires a central authority with STWS client interface for a complete communication model, to keep track of attached devices and coordinate data exchange. Virtual smart transducer from [10] is presented in terms of implementing the customized TEDS formats and additional Web services for purpose of reading/writing these TEDS. In that case, realization is limited to configuration of an output device (e.g. display) through the user-defined TEDS. On contrary, this paper describes the architectural benefits of virtual smart transducers in terms of service migration from embedded to the general purpose platform.

2.2. SOA network organization

Concept of virtual smart transducers and network data exchange via centralized server are illustrated on Fig. 1. Three types of devices are described:

a) Smart transducer is a monolithic module that integrates both: Network Capable Application Processor (NCAP) which acts as its network gateway, and Transducer Interface Module (TIM) which holds actual transducers and their TEDS.
b) NCAP and TIMs are separated and they communicate over additional IEEE 1451.X layer, where X depends on particular interface (e.g. X = 2 for serial interface, and X = 5 for wireless interface) [2].
c) From server's point of view, device appears as any other smart transducer, but it implements processing application behind the STWS interface instead of physical sensors/actuators. Each processing application (or algorithm) on device is placed to a separate logical entity named Virtual TIM (VTIM).

Central server acts as a data dispatcher between physical transducers and virtual transducers (i.e. processing applications). All devices registered in the network confirm to the same WSDL contract, hence, we attain a perfect symmetry between transducers and processing applications. During STWS Service calls, server exchanges the platform-neutral SOAP/XML messages with devices. IP for each device is obtained dynamically over DHCP protocol, which changes the common Web service usage. Web service is normally run and published on a fixed IP, while we use it as a dynamic component in the smart transducer network.

3. VTIM Architecture

In this section, we take a closer look at architectural benefits through VTIM example described on Fig. 2. VTIM accepts data via channel denoted as Channel x (virtual actuator) and produces result that it writes out to Channel x' (virtual sensor). TIMand VTIM channels that share the same name are connected by operator through a Web Manager available on the central server. Before establishing connection, operator must be sure that source and destination channels are compatible. Each device has itscopy of TEDS on the server, so that operator can have enough information about available channels.

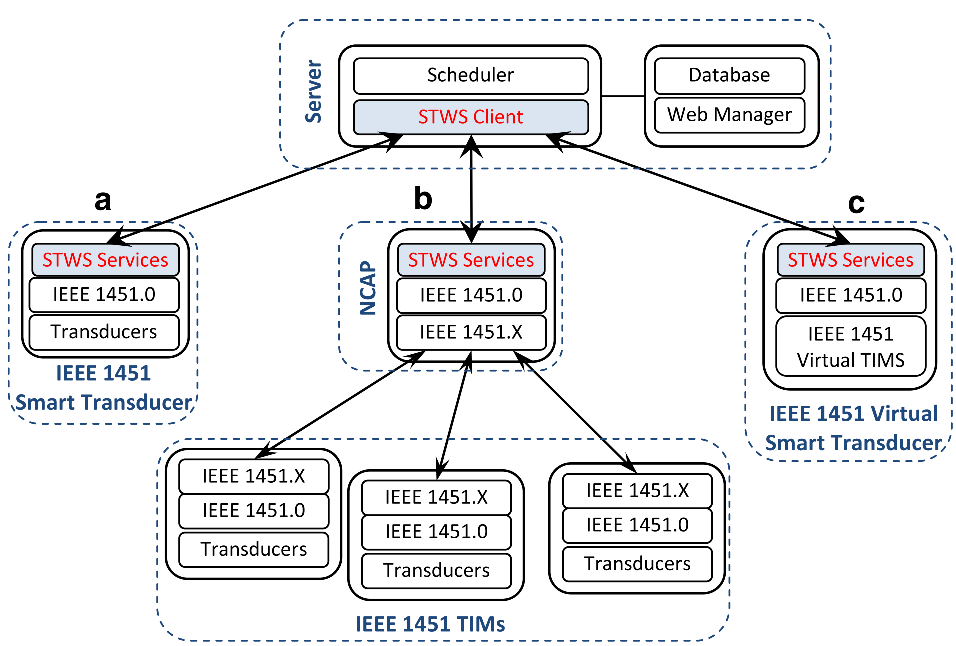

Fig. 1. Architecture overview.