Development of an OPC UA SDK Based WCF Technology and Its Deployment for Environmental Monitoring Applications

Tu Nguyen Thi Thanh and Thang Huynh Quyet

School of Information and Communication Technology,
Hanoi University of Science and Technology, Hanoi, Vietnam
thanhtu@hut.edu.vn, thanghq@soict.hut.edu.vn

Abstract. This paper focuses on a technological innovation that has been applied in many countries and still has continuously been researched to enhance its effectiveness – OPC (Openness, Productivity, and Connectivity) Unified Architecture. OPC Unified Architecture (OPC UA) is new specification used for the connection of accessories, based on open form, and independent of any technological systems. OPC UA gains access to data base and events following its real time or are stored through safe, trustworthy and totally separate connection. The research presented in this paper attempts to introduce the development of an OPC UA SDK based WCF technology and used for monitoring and control systems. The deployment of the proposed OPC UA SDK for an environmental monitoring application (EMA) is also presented as an illustration of the successful development of the proposed OPC UA SDK. This SDK in turn makes system architects and developers easy to design and implement applications in terms of environmental monitoring. In addition, this also reduces the development time and cost for such applications.

Keywords: OPC UA, SDK, XML, WCF technology, web service.

1 Introduction

Extensible Markup Language (XML) is a markup language in which users can define their own tags. XML tags not only show how the data should appear as in HTTP but also convey information about the meaning of data [1, 2]. In order to share information and knowledge among different applications, a shared set of terms describing the application domain with a common understanding is needed. XML concept has been used in the OPC XML-DA specification and the OPC UA specifications. XML based languages provide the means to represent the content, semantics, and schemata of data [3]. By utilizing the exchange of data using a non-proprietary data format, XML is particularly variable in supporting interoperability in distributed heterogeneous environments such as the Internet. XML based languages are used for developing XML based description model as studied and introduced by Wang and Lu [4], Fourer et al. [5], Nugent et al. [6], Wollchlaeger and Bangermann [7], Wollschlaeger et al. [8], etc.

P.C. Vinh et al. (Eds.): ICCASA 2012, LNICST 109, pp. 347–356, 2013.

Web services technology is software components by describing via Web Services Description Language (WSDL) in which web services are capable of being accessed via standard network protocols such as SOAP over HTTP. Today web services are accepted in practice, industry, and the business [10, 11]. The OPC Foundation now is working on supporting XML and web services as well as proposed in the OPC UA specifications [12]. The greatest advantage of web services is that they allow applications running on different platforms using different architectures and are coded by different program languages. The size of XML messages is very large compared to the size of DCOM messages. However, binary data encoding approach can be used for web services to improve the performance as pointed by Eppler et al. [14], Bayardo et al. [15], etc.

The OPC Foundation in partnership with various industry leaders is publicizing OPC, XML, web services, and openly embracing Microsoft's .NET technology. The OPC UA technology will open much of the business world to process data and distributed information. It will enable management to make informed business decisions faster. The OPC Foundation has clearly indicated that the OPC UA standard intends to enable enterprise interoperability and expects to solve enterprise integration challenges. This is a very ambitious undertaking and has been difficult to determine what elements of enterprise interoperability can actually be standardized. It is clear that the OPC UA standard does not provide everything needed for interoperability from the enterprise IT perspective, but the impact is expected to be considerable.

OPC UA has demonstrated the practicality and effectiveness of it, but in Vietnam the application and studies of OPC UA is very limited, even without any public company research results. On the other hand the specification of the organization of the OPC Foundation OPC UA standard is very flexible and not dependent on any platform, so it is very abstract and elusive. In the SDK was developed by OPC UA standard is either very expensive, not for research purposes (Prosys, UA SDK), or is not fully functional (free version of the Unified Automation), and have the general is closed for application development. Building an SDK in the specification of the OPC Foundation and write applications to illustrate the function has been supported in the SDK, then, using the SDK in the application of information management environment for validating application for practicality and efficiency of the OPC UA standard.

Nowadays, environmental pollution has become one of the hot issues of Vietnam in particular and world in general. Vietnam in the process of modernizing the country, many businesses have very high priority on the issue that returns to the disregard of sustainable development, discharge of untreated wastewater into the environment during produce serious effects on the environment that the violations were the most public concern. In order to prevent acts of unlawful discharge of environmental agencies also frequently measured and the status update in the near industrial areas, but this approach requires officials to make the observation result costly and inefficient. Therefore, the need is to setup a monitoring system/automatic quality monitoring of wastewater from factories and enterprises.

One important application of OPC UA is in the monitoring system control and data acquisition. It is optimized for communication between server and client. The OPC UA systems are used to build the central server application and client application in monitoring stations. Server will model the system's information into the address space, and the client will make the registry change tracking data to continuously update the measured data.

The research attempts to introduce the development of an OPC UA SDK and its deployment for an environmental monitoring application (EMA). This application demonstrates the successful development of the proposed OPC UA SDK. This SDK makes system architects and developers easy to develop their applications and also reduces the development time and cost for this kind of applications.

This paper is organized as follows: The next section provides a background and related work. In Section 3 introduces the design and implementation of an OPC UA SDK. In Section 4, development of an environmental monitoring application is presented. The experiment and discussion is presented in Section 5. Finally, Section 6 concludes some remarks and future works.

2 Related Work

The OPC foundation is an independent, non-profit, industry trade association comprised of more than 470 leading automation suppliers worldwide. In this section, several overviews of related OPC technologies are provided. It is dedicated to ensuring interoperability in automation by creating and maintaining open specifications that standardize the communication of acquired process data, alarms and events, historical data, and batch data to multi-vendor enterprise system and between production devices. The production devices include sensors, instruments, PLCs, RTUs, DCSs, HMIs, historians, trending subsystems, alarm subsystems, etc.

Recently, more than 22000 products are based on the OPC specifications. Current OPC specifications and those being developed include important industrial issues like Data Access (DA), Alarms and Events (AE), Historical Data Access (HDA), Security [13, 14]. The new OPC UA specifications now have been released [9, 16]. This standard is the next generation technology for secure, reliable, and interoperable transport of raw data and preprocessed information from the plant floor (or shop floors) to production planning or the ERP systems. It extends the existing OPC industry standards with fundamental features such as platform independence, scalability, high availability, Internet capability, and much more. In particular, platform independence and scalability allow completely new and cost-effective automation designs to be implemented. The OPC UA technology enables all three types of data - current data, historical data, and alarms and events - to be accessed by a single OPC server, making each type of data relative to each other. Three different types of data with different semantics today have been required in automation systems, for example, to capture the current value of a temperature sensor, an event resulting from a temperature threshold violation, and the historic mean temperature. The OPC Foundation has clearly indicated that the OPC UA standard intends to enable enterprise interoperability and expects to solve enterprise integration challenges. This is a very ambitious undertaking and has been difficult to determine what elements of enterprise interoperability can actually be standardized. It is clear that the OPC UA standard does not provide everything needed for interoperability from the enterprise IT perspective, but the impact is expected to be considerable.

3 Design and Implementation of an OPC UA SDK

To consider the situation of developing an OPC UA client-server application for specific applications, both client and server have specific functionality according to use cases, for example the OPC UA server acquires a data source from devices and the OPC UA client is used to display and draw these data. However, the problems are that trying to separate the functionality into general functionality and dependent functionality for use cases. The general functionality is classified into high-level functionality for connection management, session, etc., and lower-level functionality for data encoding, security, and data transportation, etc. Therefore, the application architecture based on the OPC UA standard can be shown in Fig. 1 (a).

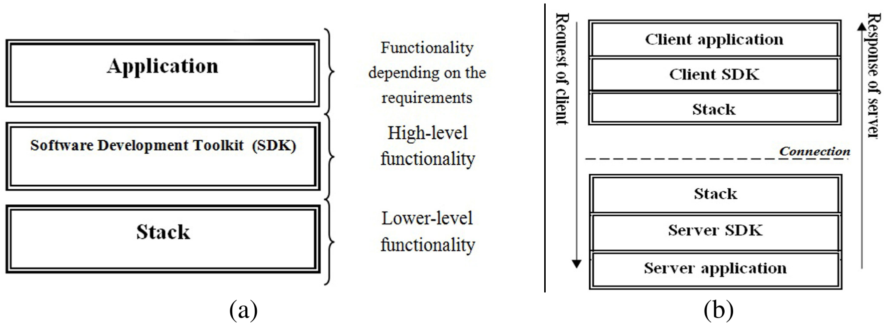

Fig. 1. (a) Application architecture diagram OPC UA standard. (b) Diagram system between server and client.

The system architecture of an OPC UA client-server application can be represented in Fig. 1 (b) including the OPC UA client application and OPC UA server application. It is easy to seem that data from the OPC UA server will pass through OPC UA Client SDK in order to convert into encoded data. The encoded data is transported by the stack in the server side. The goal of this research is only to focus on developing high-level functionality. The lower-level functionality is executed by the use of Windows Communication Foundation (WCF) technology from Microsoft.

3.1 Architecture of the Proposed OPC UA SDK

As aforementioned, the proposed OPC UA SDK has the following functionality:

i. Connection secure
ii. Session management, e.g., set timeout for the session, etc.
iii. Address space management

iv. Subscription management
v. Other functionality like logging, exception handling, etc.

3.2 The Proposed OPC UA SDK and WCF Technology

WCF technology is a platform technology in order to unify interface programming models supported in .NET 2.0. This technology allows developers to implement service solutions to reach stability, flexibility, secure-ability, etc. It reduces the development time and cost for developers [9]. Thus, WCF technology is a compatible technology for developing OPC UA standard and OPC UA SDK.

3.3 The OPC UA Server SDK

The architecture of the proposed OPC UA Server SDK with the mentioned functionality is shown in Fig. 2. The description of each component of OPC UA Server SDK is summarized as follows:

i. ServerBase: This class provides the functions of an OPC UA Server application. The server will start after the function Star() is called.
ii. IServerIntenalData: This is an interface for providing functions that will be called and used by other components in the OPC UA server. For example, SessionManager needs information from NodeManager and SubscriptionManager need to check a session managed by the SessionManager. This interface includes classes as shown like SessionManager, INodeManager, SubscriptionManager, ApplicationDescription, EndpointDescription, and ISecurityHelper.
iii. ServiceHost: This component is included in .NET Framework to create the endpoint of the services.
iv. MasterNodeManager and INodeManager: These components implement services and functions related to address space. The MasterNodeManager consist of CoreNodeManager and might be objects for other INodeManager.
v. CoreNodeManager: This component implements INodeManager Interface. When the object CoreNodeManager is created, it loads standard address space of the OPC UA server from file to memory.
vi. SessionManager: This component implements services and functions related to session management like CreateSession, ActiveSession, CloseSession, etc.
vii. Session: This component stores information for a session between the OPC UA Client application and OPC UA Server application, e.g., the function Activate allows activating a session according to the specific parameters.
viii. SubcriptionManager: This component manages subscriptions and their monitored items. This provides services and functions for subscription management such as AddSubcription, RemoveSubcription, etc.
ix. SessionPublishQueue: A session might have more subscriptions and a list of requests. This component support SubcriptionManager to choose a correspondent subscription to return Notification Messages with storing requests into queue in order to process.

x. Subcription: This component provides functions and services to manage monitored items such as AddMoniteredItem, RemoveMonitoredItem, etc.
xi. MonitoredItem: This component contains the parameters of a monitored item. The SubcriptionManager will pass data into monitored item.

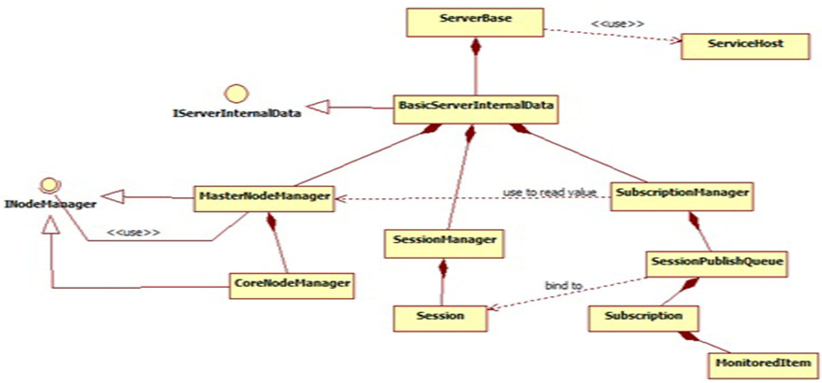

Fig. 2. The architecture of the proposed OPC UA Server SDK based WCF technology

a) Creation of Endpoint

The steps to create an Endpoint can be presented as: The OPC UA server uses ServiceHost from the .NET framework to transfer the description of an Endpoint that depends on each application. ServiceHost calls functions from class BindingFactory in the proposed OPC UA SDK to provide build the bindings for endpoint. Binding often indicates one of three kinds of bindings like WsHttpXMLBinding, UaTcpBinding, or WsHttpBinaryBinding. By the way, the object ServiceHost can create the Endpoint for the OPC UA server.

b) Initialization of the OPC UA Server

The object StandardServer in OPC UA server that provides data and its information to OPC UA client needs to be initialized all parameters. First, it loads the configuration from the file or input-parameters. Then, it initializes the object MasterNodeManager and creates the ServerObject that is a standard object in order to provide the states of the server. Finally, it starts the objects SessionManager and SubcriptionManager in order to create the Endpoint for the OPC UA server.

3.4 The OPC UA Client SDK

The architecture of the proposed OPC UA Client SDK: An OPC UA client calls services and functions that are provided by the OPC UA server by using two objects SessionClientEndpoint and DiscoveryClientEndpoint. In addition, the class ClientBase uses other components such as BindingFactory and ISecurityHelper. The BindingFactory supports the OPC UA client to have correct configurations for connecting to the OPC UA server and ISecurityHelper is used for securing the exchange information between the OPC UA client and OPC UA server.

4 Development of an Environmental Monitoring Application

As the introduction of the proposed OPC UA SDK, the development of an Environmental Monitoring Application (EMA) is implemented by using such an OPC UA SDK. This section presents (i) an overview of EMA system, (ii) devices used for the system, (iii) EMA software analysis and design and system integration.

a) Overview of system architecture

The operation of the proposed system is as follows:

i. Measurement devices placed on the rivers or lakes will measure the information about the water and this information is sent back to the OPC UA server.
ii. The OPC UA client in order to monitor such information on the devices requests latest data from the OPC UA server and draws these data in chart format to operators in real-time constraint.
iii. EMA system can alert events to operators if the mutation of water parameters that support the user to decide his/her decisions timely.

The components of the EMA system can be shown in Fig. 3. These components are divided as: data acquisition module, storage module, and workstation. Data from sensors at the data acquisition module are sent to the OPC UA server through SIM548 based on TCP/IP. The data are stored to database in which the server will provide these data to workstations, i.e., OPC UA client applications. These workstations are located in local area, which show information from sensors and devices to operators in order for representing events and suggestion of these problems.

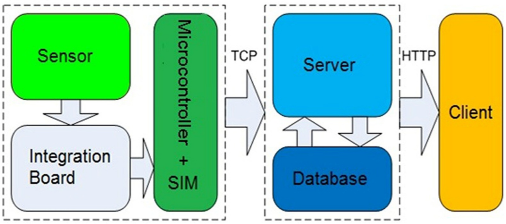

Fig. 3. The components of the EMA system

b) Devices for EMA

Sensors: The sensors are used for the EMA system as the following:

i. pH sensor (Hana, HI1230): This sensor is widely used in EMA system, which has functionality to convert the pH values into voltage values correspondingly. The voltage values of the sensor are from -414mV to +414mV (in the temperature 25oC).
ii. DO sensor (Hana, HI76409) is used to measure the concentration of dissolved oxygen in water by using Galvanic method.
iii. Conductivity sensor is two metal plates placed opposite each other in a certain volume of the water to measure the ability of conducting electricity.

iv. Light sensor (OPT101) is used to measure the light intensity at the inspection area.
v. Turbidity sensor is based on the light sensor.
vi. Depth sensor is used for indirect measurement through the use of pressure sensors by the pressure of a body of water corresponded to the depth of the water mass. The pressure sensor MPXV7035 is used that have output voltage in range from 0.2 to 4.7V.
vii. Temperature sensor LM35 is used for measuring temperature. The output voltage is compatible with the temperature of the environment.

Microcontroller and SIM: Microcontroller ATmegal128 is used because of having potential features: ADC channel with the resolution of 10 bits, two USART blocks for programming, 64 I/O registers, etc. SIM548C is used in the EMA system, which is GSM/GPRS working on 4 bands (EGSM900/DCS1800, GSM850/PCS1900). In addition, SIM supports global positioning system (GPS).

c) Software Analysis and Design

Functionality Analysis and Design: The intent of the EMA software is to acquire data that come from sensors placed at rivers and lakes in different area through GPRS and TCP/IP protocol, to store these data into database at the center. Each record is marked as a code together with device code, timestamp, etc. Based on these data, operator can make the reports and statistics for environmental situation for rivers and lakes. Form the point of view of the statistics, the operators will have evaluation and solution in order to process timely.

The functionality of the OPC UA server can be summarized as follows:

- F01S: Acquiring and storing data from measurement devices
- F02S: Real-time data query
- F03S: Report query
- F04S: Warning when parameters over threshold

The functionality of the OPC UA client can be summarized as follows:

- F05C: Device information query
- F06C: Real-time data query
- F07C: Warning display
- F08C: Report query

Database Design: The database is designed with three tables: Device table, Location table, and Measurement table. The device information is stored into Device table. Each device is unique. The Location table presents area or river or lake which devices are placed. Lastly, the Measurement table stores data coming from devices.

5 Experimental Result

The EMA (environmental monitoring application) system is basically completed for experiment in order to receiving the monitoring data from sensors to OPC UA Client application. Operators in the client side can monitor these data for their tasks. The experimental setup for environmental application and the interface of the OPC UA Client application can be shown in Fig. 4(a) and Fig. 4(b).

(a) (b)

Fig. 4. (a) The main of EMA system. (b) The hardware of EMA system.

The EMA system provides functionality for measuring parameters including pH, DO, ORP, temperature, conductivity level, depth level, etc., This information is sent to OPC UA server and then it is represented in the OPC UA client. The experimental results demonstrate the ability of the proposed OPC UA SDK for developing specific application in terms of monitoring and control. These experimental results, however, are initial results for the EMA system. For further experiment, the EMA system should be continuously developed and implemented with the accuracy of measuring parameters, high performance, and alarms and events for operators.

6 Conclusions

This paper has introduced the development of an OPC UA SDK based on the OPC UA specifications that were proposed by the OPC Foundation. The design and implementation for both sides – OPC UA Server SDK and OPC UA Client SDK is based on the Windows Communication Foundation in order to reduce the development time for the proposed SDK. The deployment of the proposed OPC UA SDK has been applied for an environmental monitoring application for acquiring the environmental data from the devices, i.e., sensors, which are placed in the rivers and lakes. This environmental monitoring application demonstrates the successful development of the proposed SDK and the ability of the SDK to real-life applications. The proposed OPC UA SDK in turn makes system architects and developers easy to develop their applications by using the developed components. It also reduces the development time and cost for general monitoring and control systems. In the future work, the system will be continuously developed for both monitoring and control functions, not only for monitoring functions until now. The developed components will be also optimized and the experiments will be conducted for providing more experimental results.

Acknowledgments. The authors would like to thank Nguyen Tri Dung for his support to implement EAM system and Dr. Vu Van Tan for his valuable comments to this paper.

References

1. Singh, M.P., Huhns, M.N.: Service Oriented Computing: Semantics, Processes, and Agents. Wiley & Sons, Chichester (2005)
2. XML Extensible Markup Language, `http://www.w3.org/XML/`
3. El Gayar, O., Tandekar, K.: An XML Based Schema Definition for Model Sharing and Reuse in a Distributed Environment. Decision Support Systems 43(3), 791–808 (2007)
4. Wang, Y.H., Lu, Y.C.: An XML Based DEVS Modeling Tool to Enhance Simulation Interoperability. In: Pro. of the 14th European Simulation Symposium, pp. 406–410 (2002)
5. Fourer, R., Ma, J., Martin, K.: OSiL – An Instance Language for Optimization, Department of Industrial Engineering and Management Sciences. Northwestern University, Chicago (2006)
6. Nugent, C.D., Mulvenna, M.D., Moelaert, F., Bergvall-Kåreborn, B., Meiland, F., Craig, D., Davies, R., Reinersmann, A., Hettinga, M., Andersson, A.-L., Dröes, R.-M., Bengtsson, J.E.: Home Based Assistive Technologies for People with Mild Dementia. In: Okadome, T., Yamazaki, T., Makhtari, M. (eds.) ICOST. LNCS, vol. 4541, pp. 63–69. Springer, Heidelberg (2007)
7. Wollchlaeger, M., Bangemann, T.: XML based Description Model as a Platform for Web based Maintenance. In: Proc. of the IEEE Conference on Industrial Informatics, pp. 125–130. IEEE Press, Los Alamitos (2004)
8. Wollschlaeger, M., Geyer, F., Krumsiek, D., Wilzeck, R.: XML Based Description Model of a Web Portal for Maintenance of Machines and Systems. In: Proceedings of the 2003 IEEE Conference on Emerging Technologies and Factory Automation, pp. 333–340. IEEE Press, Los Alamitos (2003)
9. Mahnke, W., Leitner, S.-H., Damms, M.: OPC Unified Architecture (2009)
10. Simulation. In: Proceedings of the 7th International Symposium on Distributed Simulation and Real Time Applications, p. 76. IEEE Press, Los Alamitos (2003)
11. Zeeb, E., Bobek, A., Bohn, H., Golatowski, F.: Service Oriented Architectures for Embedded Systems Using Devices Profile for Web Services. In: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, pp. 956–963. IEEE Press, Los Alamitos (2007)
12. Jammes, F., Smit, H.: Service Oriented Architectures for Devices the SIRENA View. In: Proceedings of the 3rd IEEE International Conference on Industrial Informatics, pp. 140–147 (2005)
13. The OPC Foundation: The OPC Unified Architecture Specifications: Parts 1 11, Version 1.xx (2008), `http://www.opcfoundation.org/Downloads.aspx`
14. The OPC Foundation: The OPC XML Data Access Specification. Version 1.01 5 (2004), `http://www.opcfoundation.org/Downloads.aspx`
15. Eppler, W., Beglarian, A., Chilingarian, S., Kelly, S., Hartmann, V., Gemmeke, H.: New Control System Aspects for Physical Experiments. IEEE Transactions on Nuclear Science 51(3), 482–488 (2004)
16. Van Tan, V.: A SOA Based Framework for Building Monitoring and Control Software Systems, Doctoral Dissertation. University of Ulsan, Korea (2010)