Designing and Developing a Modern Distributed Data Acquisition and Monitoring System

Vu Van Tan, Dae-Seung Yoo and Myeong-Jae Yi*

School of Computer Engineering and IT, University of Ulsan
San-29, Moogu-2 dong, Namgu, Ulsan 680-749, Republic of Korea
Email: {vvtan, ooseyds, ymj}@mail.ulsan.ac.kr; Fax: +82-52-259-1687

This paper introduces new control system aspects used for designing and implementing a distributed data acquisition and monitoring software system. These aspects are proposed according to OPC technologies and XML as standards for interfaces, functionalities, and architectures. The proposed system allows us to easily aggregate existing OPC Data Access (DA) servers and new OPC XML-DA servers into a unified and flexible system that can support complex data exchange among these OPC servers. To guarantee security of remote invocations for the proposed system including the authentication of clients, the encryption of messages, the access control, and the security level aspects are discussed. These security aspects provide more security solutions for technical-level readers. The comparison and discussion are made to indicate that the proposed system has a good design and an acceptable performance.

Keywords: Complex data; Control system aspects; Monitoring system; OPC; Security; XML.

ACM Classification: C.2.4, C.3, D.2.10, D.2.11, D.2.12, H.1.0, H.4.3, K.1.0.

1. INTRODUCTION

Web technologies today are gaining increased importance in control and monitoring systems, especially for control systems of relatively slow processes. XML now is an open standard that provides interoperability and data integration using the Internet. It is also the preferred format for encoding and moving the structured data in the independent systems. XML and Internet technologies provide new and powerful ways of assessing and delivering the plant floor data, condition monitoring and e-diagnostic to users across manufacturing enterprise using standard web-browsers and wireless devices (Holley, 2004). Therefore, the OPC Foundation formed the OPC XML-DA technical working group to define a new specification for moving the same type of plant floor data as existing OPC DA. The OPC XML-DA standard provides vertical integration between the plant floor and the condition monitoring system, maintenance system, and enterprise applications by using industrial standards, XML, and Simple Object Access Protocol (SOAP) (The OPC Foundation, 2004).

The OPC Foundation has defined the OPC Complex Data for implementing both the OPC DA and OPC XML-DA standards. The OPC Complex Data working group was making enhancements to the OPC specifications based on requirements and feedback from other industry groups to address additional data types such as structures, binary, XML, and so forth (The OPC Foundation,

* Corresponding author

Manuscript received: 9 March 2009
Communicating Editor: Zoran Salcic

2003c). Correspondingly, the design and implementation of an XML-DA server that allows the OPC clients to read and decode any type of the data from field devices were proposed and presented by Tan et al (2006). Chilingargyan and Eppler (2005) also developed a data exchange protocol based on the OPC XML-DA specification. This protocol focused on the acceptable performance by providing several extensions. Its development was motivated by achieving consistency with high level standards, multi-platform compatibility, and high performance.

To transfer data in the heterogeneous environments, the bandwidth problems need to be considered and solved. The fundamental approach to tackle such problems today is using a binary data representation with the agreement about a binary exchange format. The consideration for the memory buffer, the CPU resources, and the conversion between the XML representation and the binary representation to hold the whole data for this approach is therefore required. Moreover, only the fast possibility to transmit a large amount of data is to use the binary representation rather than the XML representation. But different platforms or even different compilers of the same platform use different binary representations such as different floating point formats, different string formats, and much more (The OPC Foundation, 2003c; Eppler *et al*, 2004). Due to them, some universal standards should be chosen to transport data between different platforms. On the other hand, the OPC XML-DA standard has a big disadvantage that used the XML textual data representation for data exchange between the client and the server. This causes much more network traffic to transfer data as discussed by Eppler *et al* (2004). That is, XML messages are very large in comparison to similar DCOM messages when carrying out the same information, i.e., the size of a SOAP message exchanged between the client and the server is very large, so that reducing the required size of an XML message in order to improve the bandwidth is needed.

Due to the development of process monitoring and control systems nowadays, it requires to aggregate hundreds or even thousands of the existing OPC DA servers and new OPC XML-DA servers into a flexible and unified system that supports the exchange of the complex data between the OPC servers. This system should be configured by using remote clients through the Internet (Tan *et al*, 2007b). For development, programmers can use the j-Interop (http://www.j-interop.org/) library to implement the proposed system in reducing the implementation cost and time. This library is a Java Open Source library for development of pure and non-native Java applications which can interoperate with any Component Object Model (COM) component (j-Interop, 2008).

The study of this paper aims at proposing and developing a unified and flexible system for the process monitoring and control system. This system is designed and developed including interfaces, functionalities, architectures, and modules. It allows to easily aggregate hundreds of the existing OPC DA servers and new XML-DA servers into a system that supports the horizontal data exchange between these servers. The intention of the proposed system is to reach to an acceptable performance in order to satisfy the requirements of industrial applications by using the binary data presentation. Moreover, the security-level aspects are also discussed to provide more information to technical-level readers.

This paper is organized as follows: The next section shows the architecture of web integration, relevant OPC techniques, and the problem statements under consideration of existing approaches. Section 3 proposes a framework for the design and implementation of Universal DA Server used for process monitoring and control systems. Section 4 investigates the database connection and the data representation between the Universal DA Server and the OPC clients according to the results of the benchmarks. In Section 5, the security level aspects for the proposed system are discussed. The comparison of the proposed system with existing approaches and the discussion are provided in Section 6. They indicate the proposed system will have a sufficient performance when applying to actual industrial environments, especially for the process monitoring and control. Finally, some conclusions and future works are marked in Section 7.

2. BACKGROUND AND PROBLEM STATEMENTS

This section describes some background on the architecture of web integration used for industrial systems and a number of the standardized specifications proposed by the OPC Foundation including the OPC Data Access, OPC XML-DA, OPC Complex Data, and OPC Data eXchange specifications. By reviewing these specifications and several related works, the problem statements are discussed.

2.1 The Architecture of Web Integration

A fundamental architecture of web integration used for industrial systems is normally consisted of three layers such as *lower layer*, *middle layer*, and *upper layer* as suggested by Wollchlaeger *et al* (2002). The lower layer provides information from automation devices to controller level of an automation system. The upper layer is based on standard IT technologies such as client-server model, using web server(s) as data source and web browsers as clients. The middle layer contains the web server(s), hosting information for the clients in the upper layer. It is composed of functionalities of business logic and performs as an application gateway between the upper level clients and the lower level automation systems, mapping the public HTTP to fieldbus protocols such as DeviceNet, ControlNet, FOUNDATION fieldbus, PROFIBUS, etc. The web server can be used to assign information from the automation and control systems to object models that can be accessed via COM/DCOM (Distributed COM) interfaces, e.g., using technologies like OPC standards to assign information from field devices to an object model. The most important problem is the clear data mapping between data and web application, because the data normally have different types and different semantic meaning. Using web technologies for monitoring and control systems of relatively slow processes requires integrating a multitude of different technologies.

2.2 Relevant OPC Specifications

The OPC Foundation (http://www.opcfoundation.org/) is an independent, non-profit, industry trade association comprised of more than 350 leading automation suppliers worldwide. In this section, several overviews of related OPC technologies are provided. Firstly, the OPC DA specification defines a set of standard COM objects, methods, and properties that specifically address interoperability requirements for the factory automation, process control, and condition monitoring applications, and much more (The OPC Foundation, 2003d; Holley, 2004). The OPC DA technology leverages DCOM allowing the client-server applications to access the plant floor via an Ethernet network distributed across the manufacturing enterprise. However, the OPC DA applications are only compatible with cooperating applications based on Microsoft Platforms.

Secondly, the OPC XML-DA specification defines a new way to move the same type of plant floor data as the existing OPC COM-DA based products (The OPC Foundation, 2004). This standard provides vertical integration between the plant floor and condition, monitoring, maintenance, and much more using XML, HTTP, SOAP, and industrial standards. It provides better connectivity and interoperability for production management and enterprise applications such as Manufacturing Execution System (MES), Enterprise Resource Planning (ERP), Enterprise Asset Management (EAM), and plant optimization that need to access the plant floor data. It is complementary with products based on the existing OPC DA specification. It was specifically designed to allow the existing OPC DA based products to be wrapped by the *OPC XML-DA*

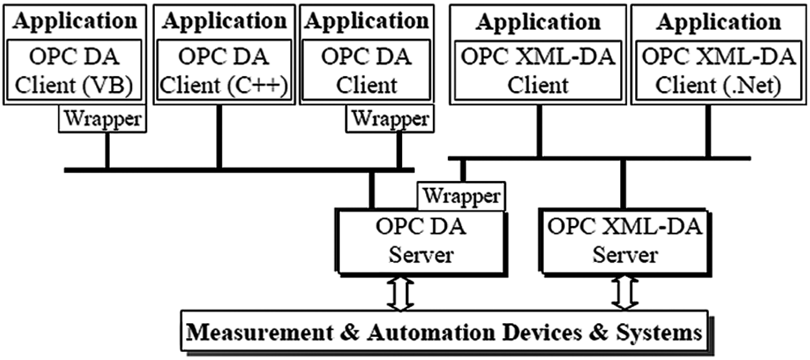

Figure 1: The OPC DA and OPC XML-DA technologies provide the plant floor to manufacturing enterprise integration

Interfaces and to effectively support both interfaces from the same OPC server. It is a standard web service interface for reading and writing data from/to the plant floor automation systems (The OPC Foundation, 2004). The OPC DA and XML-DA technologies provide plant floor to the manufacturing enterprise integration as shown in Figure 1.

Thirdly, the OPC Complex Data technology will provide a full way for the OPC clients to read and decode any type of data from field devices on the plant floor (The OPC Foundation, 2003c). Complex data mean that an OPC Item is defined as a structure. The item includes read-only information, runtime status and writeable control points. Actually, the complex data consist of complex data items that can include non-structured items, structured items, XML data, OPC binary, and so forth (The OPC Foundation, 2003c; Bustamante *et al*, 2000). By this way the OPC Complex Data specification defined two type systems that provide the levels of capability such as *XML Schema* and *OPC Binary*.

Finally, the OPC DX (Data eXchange) standard has well-defined objects that are based on the OPC COM-DA and OPC XML-DA objects (The OPC Foundation, 2003b). This standard addresses the simple mechanism for moving data between source and destination. It sets out the rules associated with the when, what, and how of moving the data between endpoints. Moreover for dealing with exceptional conditions, it defines the values to be written to the target or to be maintained at the target when good data are not available from the source. The OPC DX solution adds some key extensions by leveraging the OPC DA and OPC XML-DA standards to exchange the data between peer level OPC applications horizontally. It also extends data access to enable server-to-server data exchange during runtime and independent of the real-time applications that supported by the Ethernet networks (Hao and Hou, 2004).

2.3 Problem Statements

The OPC XML-DA solution provides more advantages as aforementioned. Based on the OPC XML-DA specification, several systems for specific industrial applications have been proposed and developed successfully (Chilingargyan and Eppler, 2005; Eppler *et al*, 2004; Katsuji *et al*, 2005; Tan *et al*, 2006; The Advosol Inc., 2008). However, the OPC XML-DA solution has a big disadvantage because of using the XML data representation (The OPC Foundation, 2004; Eppler *et al*, 2004). The XML data representation causes much more network traffic to transfer data. The OPC XML-DA standard requires XML messages to be very descriptive about the data being transferred. For example, instead of “0.555” the record <*value xsi:type = “xsd:float”*>0.555< /*value*> is sent, meaning that the bandwidth is increased about six times or even more (Eppler *et al*, 2004). Furthermore, data alignments are often required to transport data in native representation. The OPC Complex Data standard based on the use of the XML data presentation requires large amounts of memory and high intensity of memory management operations. In addition, more CPU resources and memory buffer for transformation between the native data representation and XML are required (Bustamante et al, 2000). A further issue is the present unavailability of XML versions of the OPC Historical Data Access specification (The OPC Foundation, 2003a) and the OPC Alarms and Events specification (The OPC Foundation, 2002) which are typically required in scientific experiments.

To solve the bandwidth problems, the fundamental approach to transfer data in the heterogeneous environments today is using a binary data representation, which is integrated into XML to reduce the size of a message. This approach makes a challenge to consider about the memory buffer, the CPU resources, and the conversion between the XML representation and the binary representation to hold the whole data as well as discussed by Chilingargyan and Eppler (2005). Indeed, the best possible solution to transmit the large amount of data is to use the binary representation. But different platforms or even different compilers of the same platform use different binary representations such as different floating formats, different string formats, etc. The interoperability of a system therefore makes it difficult to guarantee.

More recently, hundreds or even thousands of the existing OPC DA servers and new OPC XML-DA servers need to be aggregated into a system, which allows complex data to be exchanged not only between the OPC servers, but also between the OPC servers and the OPC clients. A commercially available system, i.e., *Universal XML-DA Server,* was developed by the Advosol Inc. (2008). This system allows that the OPC DA servers and the OPC XML-DA servers can be dynamically configured as data source for client access. Nevertheless, the performance of this system is not good because of using the XML textual data representation.

On the other hand the ability to reuse and upgrade the components of a developed system is an important factor. It will make the cost of an application reducible as well as possible. Moreover, the compatibility and interoperability of a new system with existing systems and middleware are strictly required.

In order to solve the problems mentioned above, this study proposes a distributed data acquisition and monitoring system including the following features:

1. To allow us to easily aggregate the existing OPC DA servers and new OPC XML-DA servers into a *unified and flexible* system. This system permits the data to be exchanged not only between these OPC servers, but also between the OPC servers and the OPC clients.
2. The proposed system can be configured online from local or remote clients for the configuration changes. In addition, the OPC clients can access both the COM-DA servers and the OPC XML-DA servers in a full way.
3. By using the binary data representation, the performance of the proposed system is much improved and the proposed system reaches to an acceptable performance.
4. The proposed system is purposively designed and developed including *interfaces*, *architectures*, *functionalities*, and *modules*. These components make it easy to reuse, upgrade, and maintain in order to expand new features.
5. The interoperability of the proposed system is guaranteed by the investigation of XML libraries and by using the OPC specifications. This feature makes the proposed system flexible, open, and robust.

3. DESIGNING AND IMPLEMENTING UNIVERSAL DA SERVER

3.1 The System Aspect Design

The OPC XML-DA was specially proposed to allow the existing OPC COM-DA based products to be wrapped by the *OPC XML-DA Interfaces*. A gateway application for the OPC DA servers, OPC XML-DA servers, and other gateways is required that can provide a sound base for new applications and an easy migration path for the hundreds (or even thousands) of the OPC DA products in use today. Therefore, Universal DA Server based on the OPC DX specification to allow integrating the OP DA servers, OPC XML-DA servers and gateways to horizontally exchange the complex data among them is proposed. The Universal DA Server also allows the OPC clients to access the data from field devices. This provides methods to allow the XML-DA clients to access both the OPC DA servers and the OPC XML-DA servers. Moreover, access rights can be configured such as *write-only*, *read-only*, and *read-write* from the configuration client. The roles of Universal DA Server are shown in Figure 2.

A number of OPC DA servers and OPC XML-DA servers can be configured as data sources. The data from source servers are handled according to the OPC DX standard. The Universal DA Server can be configured to allow the XML-DA clients to write data to OPC source servers by using the configuration client. The OPC complex data items of all configured servers are mapped into the address space of the Universal DA Server. The OPC XML-DA server supports the OPC clients to read and decode any type of data from the field devices (e.g., see the study of Tan *et al* (2006) for the detailed techniques and implementation). The OPC DA server was also designed and implemented for the exchange of the complex data, so that the Universal DA Server should support the OPC Complex Data specification. As a result the Universal DA Server not only solves the problems to aggregate the OPC DA servers and the OPC XML-DA servers, but also permits the OPC clients to read and decode any type of data from the field devices (Tan *et al*, 2006). Moreover, the Universal DA Server is responsible for managing the data flow on a connection from source to target. In general, it subscribes to data from the source and copies these data to the target when it is received.

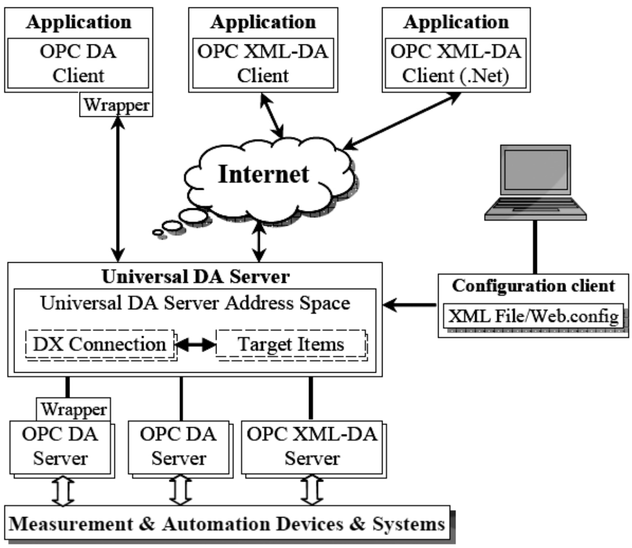

Figure 2: The proposed model of a distributed data acquisition and monitoring system

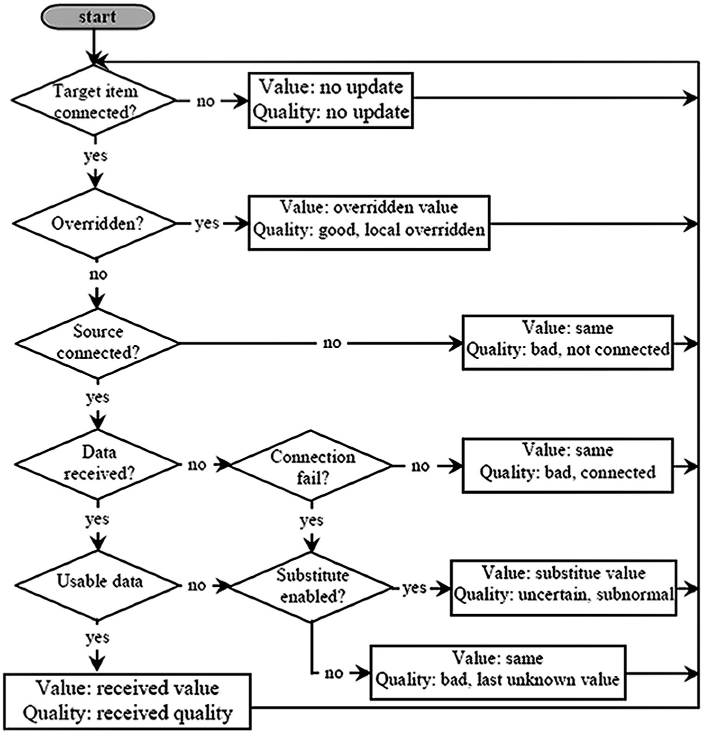

Figure 3: The flow diagram of updating data to target items

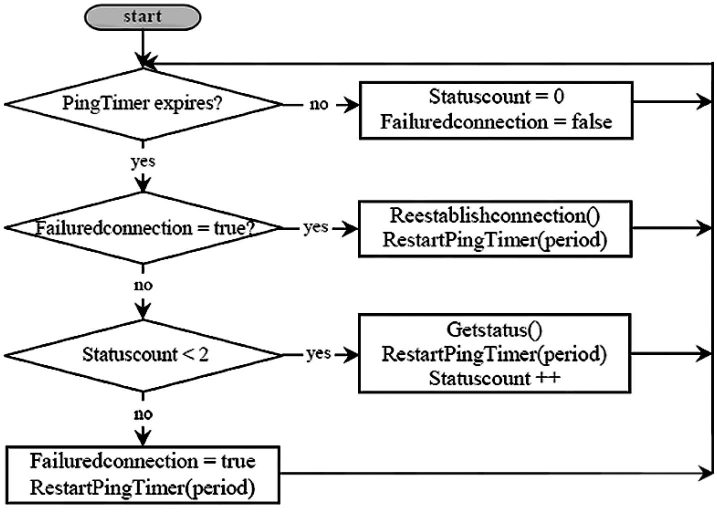

Figure 4: The flow diagram of recovering a connection failure between the Universal DA Server and the source servers

The flow diagram of updating the data from source servers to target items is shown in Figure 3. On the other hand, the Universal DA Server is responsible for maintaining the connection to source servers. When the connection to a source server is lost, either during startup or runtime, it tries to reestablish the connection in the minimum time between two attempts to reestablish a connection by using a period value, i.e., the period value can be set by the administrators or operators, e.g., 30 seconds. The flow diagram of recovering a connection failure between the Universal DA Server and the source servers is shown in Figure 4.

3.2 Modular Design of the Universal DA Server Implementation

The Universal DA Server that can access to multiple OPC DA and OPC XML-DA servers could be dynamically configured by configuration clients. The OPC items of all configured OPC servers are mapped into a single address space that can be structured in any suitable way. Both operations *Read* and *Write* are supported by the Universal DA Server. The components of the Universal DA Server for supporting integration of multiple OPC servers are shown in Figure 5.

As discussed, the Universal DA Server aggregates a large number of OPC DA servers and OPC XML-DA servers as data sources that allow the exchange of the complex data among them. The data are acquired from the source servers depending on the types of the source servers, e.g., OPC COM-DA servers or OPC XML-DA servers. In particular, the mechanism is referred to as a *Subscription*. Subscribing to the data is supported by *Callback* service in COM-DA and by *Subscribe* service in XML-DA. In the case of COM-DA, the Universal DA Server creates one or more groups for each source server. With defined groups, the Universal DA Server adds data items to them to indicate to the source server which items to access. In the case of XML-DA the Universal DA Server uses *Subscribe* and *SubscriptionPolledRefresh* services to acquire source data. The *Subscribe* service allows the clients to define a single operation for a set of the OPC items of source server to access.

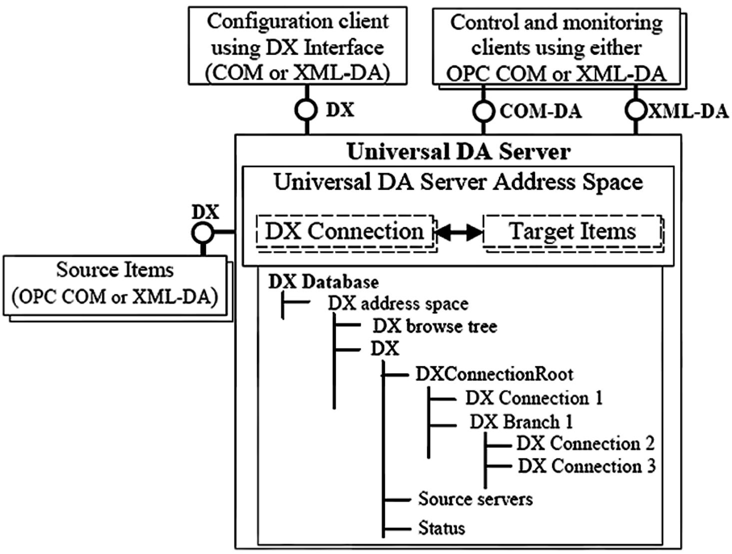

Figure 5: Components of the Universal DA Server for integrating a number of OPC DA servers and OPC XML-DA servers into a unified address space

The Universal DA Server, as Figure 5 shows, is composed of three components. The *COMModule* exposes *OPC DA-Interface* and *DX-Configuration Interface* implemented as a COM-Interface. The *SOAPModule* handles SOAP requests and exposes both the OPC XML-DA Interface and the DX-Configuration Interface implemented as a *Web Service Interface* using Web Services Description Language (WSDL) as described by Chinnici *et al* (2000). The *DXModule* contains major parts of functionality of the Universal DA Server and it is used to marshal data to/from the OPC servers. It can create new connections, remove connections, and modify the status of each connection. The Universal DA Server requires that DX-Connections can be modified and controlled by the SOAP client and the COM client simultaneously. It therefore requires solving the inconsistencies and synchronization problems.

The modules for the implementation of the Universal DA Server are shown in Figure 6. In order to support the OPC clients to read and decode any type of data from the field devices, the *OPCComplexData* class is used for providing methods to represent and convert the complex data. Complex data types are defined as dictionaries. Data types are described and presented by using the XML schema (see the approach developed by Tan *et al* (2006) for details of the implementation). Based on the OPC DX specification, the Universal DA Server provides a means to exchange the complex data between the COM-DA servers and the OPC XML-DA servers.

The *COMSOAPExchange* class used by the DXModule class aggregates the *COMModule* and the *SOAPModule*. This class handles the SOAP requests from the OPC XML-DA clients. It exchanges data between the COM-DA servers and the OPC XML-DA servers. However for all active DX Connections, the DXModule keeps up volatile runtime objects as well. It is also responsible for updating necessary status items associated with each DX Connection in order to marshal data. The status items can be used to observe the current state of the connection and data flow. The status connection information changes constantly during the runtime data exchange. As the implemented modules, the class diagram of two modules such as COMModule and SOAPModule is shown in Figure 7. These modules provide mechanisms and methods for processing the COM objects or the SOAP objects depending on the types of the OPC clients, i.e., either COM-DA clients or XML-DA clients. The SOAPModule is used not only for the XML-DA, but also for the existing COM-DA as a wrapper. It is thus composed of the *NET-XMLWrapper* and

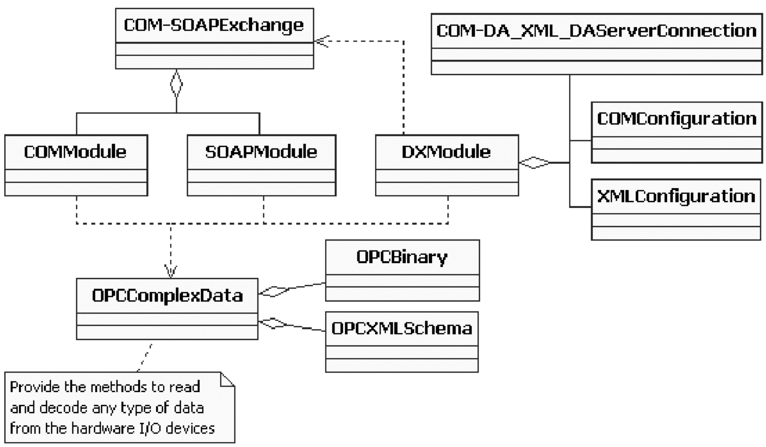

Figure 6: The UML class diagram of modules for the Universal DA Server

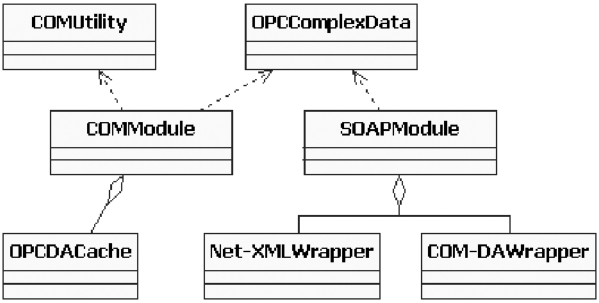

Figure 7: The UML class diagram of the COM and SOAP modules

the *COM-DAWrapper*. The *OPCDACache* provides a means to solve the large control network problems, which is aggregated in the COMModule class (Veryha, 2005).

3.3 Discussion of the Universal DA Server

The Universal DA Server provides a full way to combine hundreds or even thousands of the existing OPC DA servers and the OPC XML-DA servers into a system and to exchange the complex data between them. It allows the OPC clients to read and write the structured data from/to both the OPC DA servers and the OPC XML-DA servers. The important ability of the proposed solution is to be able to make such server deployments that allow to port web service part for non-Windows Platforms, particularly to Linux/UNIX. The Universal DA Server ensures that the OPC COM-DA components have already been developed and tested to be reused as effectively as possible. It also provides an opportunity to configure and monitor connections simultaneously using the *Web Service Interface* and the *COM Interface*. This feature is indeed used for Windows Platforms. In short, the design and implementation of the Universal DA Server guarantee the portability, the component reuse, and the ability of system extension. Furthermore, the Universal DA Server has an acceptable performance.

4. CONNECTING AND REPRESENTING DATA

This section will consider issues for connecting and representing data exchanged between the Universal DA Server and its clients. The consideration of binary data representation is introduced in Section 4.1. The comparison of XML Libraries with the results of the benchmarks is made in Section 4.2. Finally, the ability of the connection between the Universal DA Server and database for open issues when expanding the proposed system is discussed in Section 4.3.

4.1 Binary Data Representation

The best possible solution to transmit large amounts of data is to use the binary data representation rather than the XML data representation. However, different platforms or even different compilers of the same platform use different binary representations such as different byte order for representing multi-byte data, different floating point formats, etc. Thus universal standards should be chosen to transport data between different platforms. Consequently, current available binary encoding standards must be investigated and compared.

The fundamental approach to solve bandwidth problems in the heterogeneous environments is to use binary data representation, which is integrated into XML such as BXML (Bruce, 2002), BXSA (Chiu *et al*, 2005), etc. More recently, several proposals are available to satisfy these condi-

```
<ReadResponse>
  <ItemList>
    <Items ItemName="/Signal/Item" ValueType:"ArrayOfAnyType">
      <anyType xsi:type="xsdfloat">0.4554678</anyType>
      <anyType xsi:type="xsdfloat">0.5554689</anyType>
      <anyType xsi:type="xsdfloat">0.5884678</anyType>
      <anyType xsi:type="xsdfloat">0.8554678</anyType>
      <anyType xsi:type="xsdfloat">0.9559679</anyType>
      <anyType xsi:type="xsdfloat">0.9994678</anyType>
      <anyType xsi:type="xsdfloat">1.4554678</anyType>
      <anyType xsi:type="xsdfloat">1.8554678</anyType>
    </Items>
  </ItemList>
</ReadResponse>                                              (a)
------------------------------------------------------------------------------
<ReadResponse>
  <ItemList>
    <Items ItemPath="/S/Float" ValueType:"BinaryStream">
      <BinaryStream type="typename" xlink=cid:SP1>
    </Items>
  </ItemList>
</ReadResponse>
----
Content-ID: SP1
...32 bytes BinaryData                                       (b)
```

Figure 8: (a) A set of eight measurement values presenting in XML.
(b) Replacement of these eight values by an XLink reference with binary data attachment.
The required size of a SOAP message with the attachment of the binary data is significantly reduced.

tions such as the SOAP message with attachments (SOAP, 2003) or the HTTP message using XLink (XML Linking Language) suggested by DeRose *et al* (2001). The Universal DA Server will respond to the XML-DA clients with the SOAP message with all the data replaced by XLink references. To incorporate binary data into the SOAP message, the WS-attachment technologies (Barton *et al*, 2000; Gailey, 2003; Freisleben *et al*, 2004) could be used. On the other hand XLink is used to link different parts of a compound message together. Also, addressing a multi-group can be done in XLink. Figure 8 illustrates the use of the binary data representation with the WS-attachment technology in the SOAP message in order to reduce the required size of a message, i.e., the SOAP header is still XML to pass firewall and the body of the SOAP message and should be encoded as binary data.

4.2 Comparison of XML Libraries

When designing and developing an application, the problems of the existing approaches have to be solved. If there is new software available made by different manufacturers, the best things to do are benchmarks. To achieve a better basic for a decision to select the best XML library, several benchmarks were performed to evaluate the present XML libraries. Up to date, there are many available XML library benchmarking projects such as XMark (2002), XML Benchmark (2002), SAX Parser Benchmark (SAXPB, 2004), and much more (e.g., see the research studied by Mlýnková (2008) for more recent XML benchmarking projects from the point of view of various applications). However, they are not very applicable for the OPC XML-DA solution. In addition, they tested only one or two aspects of XML processing with some predefined sequences of XML data.

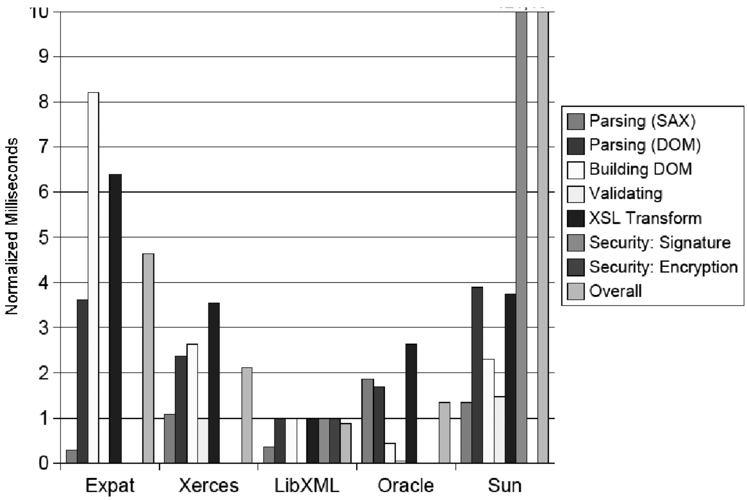

Figure 9: The significant benchmark results of five different library combinations. The right-most bar shows the overall performance of all processing stages. A blank bar indicates that the service is not available.

To provide fast and reliable Universal DA Server solution, a fast multiplatform XML toolkit is required to process different types of XML files at high speed. Also, some supports including XML-binary Optimized Packing (Gudgin *et al*, 2005), XML Encryption (Eastlake and Reagle, 2002), XML Signature (Boyer *et al,* 2002; Eastlake *et al*, 2002), XML Transformation (Clark, 2001), and some other should be investigated. The significant benchmark results of five different libraries are shown in Figure 9 (e.g., see the study introduced by Chilingarian (2003) for more details about the setup and the size of data-set used for benchmarking). In practice, multiple compilers are available and they should be compared because the performance of the XML libraries is very important for the overall system performance. Figure 9 shows the overall performance of all processing stages in the right most bar. The performance tests can be divided into the following phases of XML processing stages such as Schema validation, XSL Transformation, XML Security, XML Parsing, etc. As a consequence, the *LibXML* is effectively high performance and it is selected to apply to the proposed system.

4.3 Connection between the Universal DA Server and Database

To provide an open and flexible system for implementing and aggregating the OPC HDA (The OPC Foundation, 2003a) into the proposed system for future development, the connection between the Universal DA Server and database is investigated and provided. Standard SQL (Structure Query Language) as basic query language for database searches in many cases is not efficient. Consequently, each database has its own extension to SQL such as PL/SQL for Oracle, PL/pgSQL for postage-SQL. However, SQL solutions are usually not platform independent. The platform independence can be achieved by using the XQuery language (XQuery, 2007). But in large applications the XQuery language is still too slow and too memory consuming as shown in Figure 10 (Eppler *et al*, 2004). When the Universal DA Server is defined as a web server and supports storage and exchange of the complex data between the server and the OPC clients, the connection between the Universal DA Server and database should be designed for further extension. To build a full system based on the OPC specifications, the ability to extend and integrate the proposed

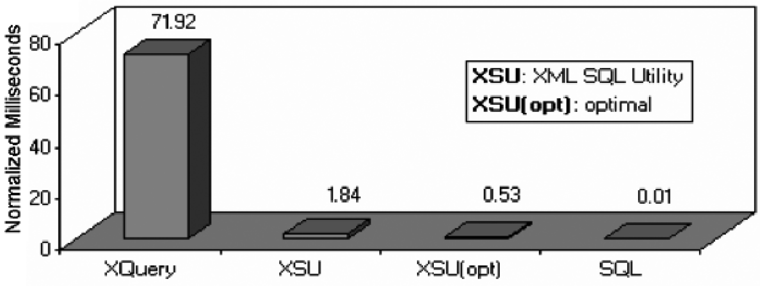

Figure 10: The comparison between query languages. SQL is obviously the fastest access method. However, XSU is feasible for querying in platform independent databases.

system when adding new OPC products or combining the existing OPC-based products should be guaranteed satisfactorily.

The situation of the XQuery language will change rapidly in the near future due to efforts put into the further development of XML. The solution to tackle this development of XML should be started with the product depending query language XSU and to migrate to XQuery (XQuery, 2007; Su and Liao, 2008; Eppler *et al*, 2004).

5. SECURITY CONSIDERATION

Security to remote invocations in the heterogeneous environments (distributed systems) is an important issue including the authentication of clients, the encryption of messages, and the access control. To achieve the required security criteria including the authentication of clients, the encryption of messages, and the access control, the concepts and solutions developed for generic IT systems have to be considered and applied. That is, appropriate technologies like encryption, Secure Socket Layer (SSL) technology, HTTP-S (Secure HTTP), certificates and digital signatures should be used (Dzung *et al*, 2005; Wollchlaeger *et al*, 2002; Gutiérrez *et al*, 2006). The security solution for the proposed system has to guarantee the security for both the COM-DA and XML-DA products. In addition to reducing the bandwidth problems, the described solutions of improving performance raise new security problems. In the standard cases, the HTTP-S can be used to protect data. However for multicasting data connections, HTTP-S is not available and some other mechanisms must be used. Besides, a proposal is to use an authentication server that will use SSL private/public keys for authorization and generate symmetric session keys. It is also proposed to use the internal XML security approach to the control connections instead of the HTTP-S protocol, being described in XML Encryption (Eastlake and Reagle, 2002), XML Signature specifications (Boyer *et al*, 2002; Eastlake *et al*, 2002), and XML Decryption (Hughes *et al*, 2002).

For the COM-DA products the security based on DCOM security that distinguished between connection security, per call security, and per packet security was proposed and developed (The OPC Foundation, 2000; Dzung *et al*, 2005). Moreover, access control and launch permission can be set for individual DCOM objects, groups of objects, or all objects on the OPC servers. They are used for security in the COM-DA Interface. Other aspects for security guarantees to the monitoring and control systems including the guarantee of both the Intra/Internet and the field-device level were proposed and presented by Treytl *et al* (2005).

Security objectives to remote invocations in distributed environments will include *integrity*, *confidentiality*, *availability*, *authentication*, *non-repudiation*, *accountability*, and *reliability* (Singh and Huhns, 2005; Dzung *et al*, 2005). The XML signature surely meets the requirements of the security objectives like integrity, non-repudiation, and authenticity. Thereby, it also implicitly satisfies the security objectives accountability and confidentiality due to the authentication. Therefore, digital signatures are suitable for a modern distributed data acquisition and monitoring system that is used for process and factory automation in the heterogeneous environments. The XML distributed signature approach (Miyauchi, 2005) is preferred to guarantee the security with the client-server model because this solution optimally reduces the data transfer between the server and the client, and supports thin clients, i.e., it is not necessary to build a special environment for XML signature creation on the clients' side.

The security for the proposed system has to prevent the attack from Ethernet and Internet environments. Several security solutions used for the application domains of OPC-based monitoring and control systems have been proposed and presented by Tan *et al* (2007a). So far, several possible security solutions have been considerately discussed. In this study, the optimal approach based on the XML distributed signature approach (Tan *et al*, 2007a), which was modified from a distributed signature approach proposed by Miyauchi (2005), is applied for the Universal DA Server. It reduces the data transfer between the client and the Universal DA Server. XML signature processing steps are divided for the server and the client to ensure that the server and the client cooperate to perform XML signature processing. Processing performed on the client is limited to the creation of the signature value based on digest value encryption using a secret key and the acquisition of a public key certificate. Thereby, the amount of data transferred is extremely small because the processing steps are distributed to the server. Complying with the security consideration above, the modified security solution can satisfy the security objectives as aforementioned.

6. COMPARISON AND DISCUSSION

In this section, the evaluation of the proposed system by comparing with the existing approaches and by discussing the useful roles of the Universal DA Server in industrial systems today is analyzed and presented. A number of the advantages of the proposed system are also provided.

6.1 Comparison with Existing Approaches

The comparison of the proposed system with other approaches is difficult due to the conceptual nature, different architectures, and wide range of production environments. Therefore, a qualitative comparison of the proposed system and the existing approaches could be made. The structural characteristics are used to compare with the existing systems: (i) Universal XML-DA Server – a commercially available system developed by the Advosol – (The Advosol Inc., 2008), (ii) KATRIN Slow Control System (Eppler et al, 2004), and (iii) another system presented by Chilingargyan and Eppler (2005). The proposed system has a number of advantages as follows:

1. The proposed system allows us to easily aggregate the existing OPC DA servers and new OPC XML-DA servers into a *unified and flexible* system in which the complex data can be exchanged not only between the OPC servers and the OPC clients, but also between these OPC servers, i.e., DA servers and XML-DA servers. In practice, a commercially available system developed by the Advosol Inc. (2008) also has the same functionality, but the performance of this system is weak because of using the XML textual data representation.
2. By using both the binary data representation in SOAP messages and the optimal XML library, the required size of a SOAP message is reduced about six times or even more compared to the size of a pure SOAP message. Thus bandwidth of the network and system resources are truly improved as much as possible.

3. The proposed system can be configured online from local or remote client that makes it flexible, optimal, and robust. A number of OPC DA servers and XML-DA servers can be configured as data sources for the client access according to the configured access rights such as *write-only*, *read-only*, and *read-write* while this feature cannot be executed by the KATRIN Slow Control System (Eppler *et al*, 2004) or another system developed by Chilingargyan and Eppler (2005).
4. The proposed system is designed and developed in a systematic way including *architecture*, *interfaces*, *functionalities*, and *modules*. By this way these components in terms of framework component and software engineering can be effectively reused and maintained in order to apply for the further development of other specific applications where the control is distributed and large amounts of distributed data are collected. The design solution implemented within the proposed system ideally reaches to an ultimate mechanism for service developers and technical-level readers. Thereby, the components of this solution are easy to identify and they will be reused for the future development of modern automation software.
5. The ability of the proposed system is a possible extension for adding new OPC-based products. Therefore, the development of OPC systems according to the OPC DA specification (The OPC Foundation, 2003d), OPC XML-DA specification (The OPC Foundation, 2004), OPC HDA specification (The OPC Foundation, 2003a), and the OPC AE specification (The OPC Foundation, 2002) can be incorporated into the proposed system in a systematic and efficient way. The connection between the Universal DA Server and database was investigated for the further development of the OPC HDA standard (The OPC Foundation, 2003a). Furthermore, the overall security aspects presented in this study provide more security solutions to technical-level readers.

6.2 Discussion

The proposed system, simply called *Universal DA Server*, was proposed, designed, and developed including architectures, interfaces, functionalities, and modules as well as an open approach to the further developments of modern automation software. As a consequence, this study provides information about the design and implementation of a process control and monitoring system based on the OPC technologies for application developers and programmers. This system provides fully a way to combine the hundreds (or thousands) of the existing OPC servers into a unified and flexible system that can be configured by the remote or local clients based on the XML-DA Interface or the COM-DA Interface. The Universal DA Server provides an optimal mechanism to aggregate the existing OPC DA servers and new OPC XML-DA servers, allowing the exchanges of complex data between these OPC servers, i.e., *COM-DA servers* and *XML-DA servers*.

The ability to run on Microsoft Platforms and non-Microsoft Platforms indicates the proposed system is very flexible for applying to the Internet environments. This means that the interoperability of the proposed system is compatible with different operating systems and middleware. The data transferred over the Internet are based on the XML technology with the attachment of the binary data representation for the SOAP message to improve the system performance. Unfortunately, the possibility of obtaining the real-time data based on the proposed system is difficult to guarantee because real-time applications must receive timely data updates in packages that are small enough that they can be digested quickly. However, business applications such as ERP and EAM systems require updates that are at hourly frequencies at best or every several minutes, and usually update every shift or more. Remote monitoring applications such as web-based process visualization normally require very slow update rates. Therefore, the proposed system has a sufficiently good performance and can be accepted for deploying to real industrial applications in future works.

The high speed Ethernets, e.g., 100Mbps and 1Gbps, gradually become common in Local Area Network (LAN) and Wide Area Network (WAN) due to the network architecture. Additionally, the bandwidth resources are increasing fast because of the development of switch technologies. The system performance will be improved as much as possible. The most important advantage of the proposed system is to allow aggregating the OPC DA servers and the OPC XML-DA servers into the system for supporting the horizontal exchanges of the complex data. Moreover, the Universal DA Server's performance is bettered by using the attachment of the binary data representation in the SOAP message when the complex data are transferred over the Internet. Finally, the proposed system is designed and developed as a framework as well as an open approach for further extension. In fact, this feature makes it easy for deploying, reusing, and customizing the proposed system in order for applying to the OPC-based industrial applications in terms of software engineering.

The system developed is being investigated about the interoperability, acceptable performance, and the ability of deployment before applying to industrial systems with a very large set of data and a short period of time.

7. CONCLUDING REMARKS AND FUTURE WORKS

This paper has introduced the design and implementation of a modern distributed data acquisition and monitoring system based on the OPC specifications and relevant IT standards. The proposed system allows the existing OPC DA servers and the OPC XML-DA servers to be aggregated into a system that permits complex data exchange between the OPC servers, i.e., COM-DA servers and XML-DA servers. This solution is to provide a sound base for new applications and an easy migration path for the hundreds (or even thousands) of the existing OPC DA servers in use today. Aiming at overcoming the performance limitations of the XML textual representation, the binary data representation exchanged between the Universal DA Server and the clients was proposed to use in achieving a sufficient good performance.

The work presented in this paper has three main contributions: Firstly, the Universal UA Server was proposed that allows a number of OPC DA servers and new OPC XML-DA servers to be integrated into a unified and flexible system, which supports the complex data exchange not only between the OPC servers, but also between the OPC servers and clients. The proposed system can be configured online using remote or local client that makes it flexible and optimal. Secondly, the XML Libraries were investigated and compared in order to choose XML technology as a candidate for the proposed system. The *LibXML* was used for solving the system requirements of monitoring and control systems today, e.g., *any platform*, *acceptable performance*, etc. Therefore, the proposed system has a sufficiently good performance and it will be accepted to apply to real industrial applications in the future. The third contribution is the security-level aspects that were discussed adequately to use good security solutions. These aspects provided seamless security solutions for the proposed system that is communicated using the Internet environments. They also provided more security information to technical-level readers. Furthermore, several open issues for the Universal DA Server were provided to develop modern distributed data acquisition and monitoring systems with collaboration of other OPC specifications-based products.

In terms of software engineering and framework component, the proposed system was designed and developed including architectures, interfaces, and modules as well as an open approach to reach to an ultimate solution in designing and developing process and factory automation software. These components will be reused for the developments of process and factory automation where the control is distributed and large amounts of distributed data are collected.

The evaluation and the discussion of the proposed system were carefully analyzed and presented.

They indicated the proposed system has a sufficiently good performance and is feasible to apply for control and monitoring systems. Because the system performance of the proposed system is dependent on the size of a SOAP message transmitting on the network, therefore, a study of the performance of the proposed system related to the size of the SOAP message with the attachment of binary data encoding will be investigated in future works before deploying the proposed system to real industrial monitoring and control applications.

ACKNOWLEDGEMENTS

This work was supported in part by the Korean Ministry of Knowledge Economy and Ulsan Metropolitan City through the Network-based Automation Research Center (NARC) at the University of Ulsan and by the Korea Research Foundation Grant funded by the Korean Government (NRF-2009-0076248). The authors also would like to thank the Editor-in-Chief, Professor Sidney A. Morris, and the anonymous referees for their valuable comments and suggestions that helped us to improve this work.

REFERENCES

BARTON, J.J, THATTLE, S. and NIELSEN, H.F. (2000): SOAP message with attachments. W3C Recommendation, http://www.w3.org/TR/SOAP-attachments. Accessed August 2006.

BOYER, J., HUGHES, M. and REAGLE, J. (2002): XML-Signature XPath filter, Version 2.0., W3C Recommendation, http://www.w3.org/TR/2002/REC-xmldsig-filter2-20021108/. Accessed November 2006.

BRUCE, C.S. (2002): Cubewerx position paper for binary XML encoding, http://www.cubewerx.com/main/HTML/Binary_XML_Encoding.html. Accessed November 2006.

BUSTAMANTE, F., EISENHAUER, G., SCHWAN, K. and WIDERNER, P. (2000): Efficient wire formats for high performance computing. In *Proceedings of the ACM/IEEE Conference on Super Computing*. IEEE Press, Los Alamitos, 64.

CHILINGARYAN, S. and EPPLER, W. (2005): High speed data exchange protocol for modern distributed data acquisition systems based on OPC XML-DA. In *Proceedings of the 14th IEEE-NPSS Real-time Conference*. IEEE Press, Los Alamitos, 352-356.

CHILINGARIA, S. (2003): Fast, multiplatform XML toolkits comparison: Functionality, speed and memory usage. http://prdownloads.sourceforge.net/xmlbench/features.pdf, http://prdownloads.sourceforge.net/xmlbench/benchmark.pdf. Accessed October 2006.

CHINNICI, R., MOREAU, J.J., RYMAN, A. and WEERAWARANA, S. (2000): WSDL – Web services description language, Version 2.0., W3C Recommendation, http://www.w3.org/TR/wsdl20/. Accessed August 2006.

CHIU, K., DEVADITHYA, T., LU, W. and SLOMINSKI, A. (2005): A binary XML for scientific. In *Proceedings of the 1st International Conference on e-Science and Grid Computing*, 336-343.

CLARK, J. (2001): XSL transformation (XSLT), Version 1.1. W3C Recommendation, http://www.w3.org/TR/2001/WD-xslt11-20010824/, Accessed August 2006.

DEROSE, S., MALER, E. and ORCHARD, D. (2001): XLink (XML Linking Language), Version 1.0, W3C Recommendation, http://www.w3.org/TR/xlink/. Accessed August 2006.

DZUNG, D., NAEDELE, M., HOFF, T.P.V. and CREVATIN, M. (2005): Security for industrial communication systems. In *Proceedings of IEEE*, 93(6): 1152-1177.

EASTLAKE, D. and REAGLE, J. (2002): XML encryption syntax and processing. W3C Recommendation, http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/. Accessed August 2006.

EASTLAKE, D., REAGLE, J. and SOLO, D. (2002): XML signature syntax and processing. W3C Recommendation, http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/. Accessed August 2006.

EPPLER, W., BEGLARIAN, A., CHILINGARIAN, S., KELLY, S., HARTMANN, V. and GEMMEKE, H. (2004): New control system aspects for physical experiments. *IEEE Transactions on Nuclear Science*, 51(3): 482-488.

FREISLEBEN, B., FRIESE, T., HEINZL, S., MATHES, M. and SMITH, M. (2006): Flex-SwA: Flexible exchange of binary data based on SOAP messages with attachments. In *Proceedings of the IEEE International Conference on Web Services*. IEEE Press, Los Alamitos, 3-10.

GAILEY, J.H. (2003): Using web services enhancements to send SOAP messages with attachments. Microsoft Documents, http://msdn2.microsoft.com/en-us/library/ms996944.aspx. Accessed May 2007.

GUDGIN, M., NENDESOHN, N., NOTTINGHAM, M. and REULLAN, H. (2005): XML-binary optimized packing. W3C Recommendation, http://www.w3.org/TR/2005/REC-xop10-20050125/. Accessed May 2007.

GUTIÉRREZ, C., FERNÁNDEZ-MEDINA, E. and PIATTINI, M. (2006): Towards a process for web services security. *Journal of Research and Practice in Information Technology*, 38(1): 57-67.

HAO, X. and HOU, S. (2004): OPC DX and industrial ethernet glues fieldbus together. In *Proceedings of the Conference on Control, Automation, Robotics, and Vision*. IEEE Press, Los Alamitos, 1: 562-567.

HOLLEY, D.W. (2004): Understanding and using OPC for maintenance and reliability applications. *IEE Computing and Control Engineering*, 28-31.

HUGHES, M., IMAMURA, T. and MARUAMA, H. (2002): Decryption transformation for XML signature. W3C Recommendation, http://www.w3.org/TR/2002/REC-xmlenc-decrypt-20021210/. Accessed March 2007.

JIA, Z. and LI, X. (2005): OPC-based architecture of embedded web server. In WU, Z., CHEN, C., GUO, M. and BU, J. (eds.) ICESS 2004, *Lecture Notes in Computer Science*, Vol. 3605, Springer, Berlin, 362-367.

J-INTEROP (2008): Pure Java-COM Bridge: j-Interop - Java Open Source Library, Version 2.05, http://www.j-interop.org/. Accessed February 2009.

KATSUJI, U., SHIN-ICHI, S. and HIDEHIKO, W. (2006): A prototype embedded XML-DA server and its evaluations. In *Proceedings of the SICE-ICASE International Joint Conference*, 4331-4336.

MIYAUCHI (2005): XML Signature/Encryption – The basic of web service security. *NEC Journal of Advanced Technology*, 2(1): 35-39.

MLÝNKOVÁ, I. (2008): XML benchmarking: Limitations and opportunities. Technical Report, Department of Software Engineering, Charles University, Czech Republic, http://www.ksi.mff.cuni.cz/~mlynkova/doc/tr2008-1.pdf. Accessed February 2009.

SAXPB (2004): SAX parsers benchmark, http://piccolo.sourceforce.net/bench.html. Accessed February 2007

SINGH, M.P. and HUHNS, M.N. (2005): *Service-Oriented Computing – Semantics, Processes*, Agents. John Wiley & Sons, Chichester.

SOAP (2003): SOAP – Simple Object Access Protocol, Version 1.2, http://www.w3.org/TR/SOAP. Accessed February 2006.

SU, H. and LIAO, H. (2008): XQPE: A partial evaluation system for XQuery. In *Proceedings of the annual IEEE International Computer Software and Application Conference*. IEEE Press, Los Alamitos, 347-354.

TAN, V.V., YOO, D.S. and YI, M.J. (2006): Design and implementation of web service by using OPC XML-DA and OPC complex data for automation and control systems. In *Proceedings of the 6th IEEE International Conference on Computer and Information Technology*. IEEE Press, Los Alamitos, 63.

TAN, V.V., YOO, D.S. and YI, M.J. (2007a): Security in automation and control systems based on OPC techniques. In *Proceedings of the 2nd International Forum on Strategic Technology*. IEEE Press, Los Alamitos, 136-140.

TAN, V.V., YOO, D.S. and YI, M.J. (2007b): Modern distributed data acquisition and control systems based on OPC techniques. In *Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, Arizona, USA. IEEE Press, Los Alamitos, 115-122.

THE ADVOSOL, INC. (2008): Universal XML-DA server, Version 1.01, http://www.advosol.com/c-5-xml-da-gateways.aspx. Accessed March 2008.

THE OPC FOUNDATION (2000): The OPC security custom interface specification, Version 1.0, http://www.opcfoundation.org/Downloads.aspx. Accessed February 2006.

THE OPC FOUNDATION (2002): The OPC alarms and events access specification, Version 1.0, http://www.opcfoundation.org/Downloads.aspx. Accessed February 2006.

THE OPC FOUNDATION (2003a): The OPC historical data access specification, Version 1.0, http://www.opcfoundation.org/Downloads.aspx. Accessed February 2006.

THE OPC FOUNDATION (2003b): The OPC data eXchange specification, Version 1.0, http://www.opcfoundation.org/Downloads.aspx. Accessed February 2006.

THE OPC FOUNDATION (2003c): The OPC complex data specification, Version 1.0, http://www.opcfoundation.org/Downloads.aspx. Accessed February 2006.

THE OPC FOUNDATION (2003d): The OPC data access specification, Version 3.0, http://www.opcfoundation.org/Downloads.aspx. Accessed December 2005.

THE OPC FOUNDATION (2004): The OPC XML data access specification, Version 1.01, http://www.opcfoundation.org/Downloads.aspx. Accessed December 2005.

TREYTL, A., SAUTER, T. and SCHWAIGER, C. (2005): Security measures in automation systems: A practice-oriented approach. In *Proceedings of the 10th IEEE Conference on Emerging Technologies and Factory Automation*. IEEE Press, Los Alamitos, 2: 847-855.

VERYHA, Y. (2005): Going beyond performance limitations of OPC DA implementation. In *Proceedings of the 10th IEEE Conference on Emerging Technology and Factory Automation*. IEEE Press, Los Alamitos, 1: 47-53.

WOLLCHLAEGER, M., NEURMANN, P. and BANGEMANN, T. (2002): Web service for remote maintenance of Fieldbus based automation systems. In *Proceedings of the IEEE International Conference in Africa*. IEEE Press, Los Alamitos, 247-252.

WOLLCHLAEGER, M. and BANGEMANN, T. (2004): XML based description model as a platform for web-based maintenance. In *Proceedings of the IEEE Conference on Industrial Informatics*. IEEE Press, Los Alamitos, 125-130.

XMARK (2002): XMark – An XML benchmark project, http://monetdb.cwi.nl/xml/index.html. Accessed December 2006.

XMLBENCHMARK (2002): XML benchmark http://www.sosnoski.com/opensrc/xmlbench/download.html. Accessed December 2005.

XQUERY (2007): XQuery 1.0 – An XML Query language, http://www.w3.org/TR/xquery/. Accessed February 2009.

BIOGRAPHICAL NOTES


Vu Van Tan

Vu Van Tan was born in Haiduong Province, Vietnam, in 1981. He received his Eng. degree in Information Technology from the Hanoi University of Technology, Vietnam, in 2004. He also received his PhD degree in Information Technology from the University of Ulsan, Republic of Korea, in 2010. He has worked as a design and analysis engineer in KhaiTri Software Company, Vietnam, for one year (June 2004 – August 2005). He is currently a post-doctoral researcher at the Network Based Automation Research Center (NARC), University of Ulsan. His main interests are software engineering, software for automation systems, internet technologies for industrial automation systems, agent technologies for process monitoring and control, and real-time communication systems.


Dae-Seung Yoo

Dae-Seung Yoo received the BS and MS degrees in Computer Engineering and Information Technology from the University of Ulsan, Republic of Korea, in 1998 and 2001 respectively. He was a guest professor in the School of Computer Engineering and Information Technology, University of Ulsan. He now works as a senior member of the engineering staff at the Electronics and Telecommunications Research Institute (ETRI). He is a member of KIISE, KIPS, and IEICE. His main interests are software engineering, software for automation systems, and Internet technologies for industrial automation systems.


Myeong-Jae Yi

Professor Myeong-Jae Yi received his BS degree in Computer Science from the Seoul National University, Republic of Korea, in 1987. He also received the MS and PhD degrees in Computer Science from the Seoul National University in 1989 and 1995, respectively. He was a part-time lecturer at the Department of Computer Science of the Seoul National University and the Sookmyung Women's University from 1991 to 1996. He is currently a professor in the School of Computer Engineering and Information Technology, University of Ulsan, Republic of Korea. Prof. Yi is also a deputy-director of NARC (Network based Automation Research Center) at the University of Ulsan and is a member of KIISE and KIPS. His main interests are software engineering, software for automation systems, Internet technologies for industrial automation systems, mobile agent, and E-commerce.