Industrial Processes Supervision Using Service Oriented Architecture

Francisco Sales de L. Filho, Aduato L. T. B. da Fonseca and Alessandro J. de Souza
CEFET
Natal-Brazil
Email: adauto6@netscape.net
{salesfilho,ajdsouza}@cefetrn.br

Felipe Alves Couto, Rodrigo P. R. dos Santos and Luiz Affonso Guedes
UFRN
Natal - Brazil
Email: lipecouto@gmail.com
{rodrigop,affonso}@dca.ufrn.br

Abstract—Nowadays the great challenge in an industrial environment is the possibility of integration of the diverse technologies in one plant of industrial automation. In this context, this article discusses the possibilities of the use of a service-oriented architecture to solve problems of interoperability between the various systems of industrial automation process.

I. INTRODUCTION

Along the years, there has been a steady increase in the automation of the industrial process. This is reflected on increase in productivity and on the improvement in the quality of the products. However, the rapid growth of industrial automation created a substantial number of equipment each with its specific way of communication. The various devices (sensors, actuators) involved in an industrial network brought to attention the reality of the applications involved in an area of industrial automation. Thus, the challenge to standardise the communication between the various process.

Software solutions based on an architecture of distributed objects like Distributed Component Object Model-DCOM, Remote Method Invocation-RMI, and Common Object Request Broker Architecture-CORBA were proposed and accepted by the industrial community and are used up to the present. Professionals in the area of automation still face the challenge of refining communication between industrial devices.

In this context, this article proposes the use of an architecture oriented to services (a services-oriented architecture) as an alternative to present standards, providing characteristics like interoperability, flexibility, portability and scalability.

This article was written so as to present concepts on Communication in the Industrial Environment, Web Service and its basic technologies, as can be seen in sections 2 and 3, and to display the results attained as described in section 4. To obtain the results, we have made a case study where we use a Web Service with the application written in Java language to manage the PLC SIMATIC S7-200 of SIEMENS.

II. COMMUNICATION IN INDUSTRIAL ENVIRONMENT

The communication of data in the industrial environment became a necessity from the beginning of process automation, where there was the need to transport to the controllers signals coming from the instruments. Since then, it has been possible to observe the evolution of these means of transport such as pneumatic, hydraulic, electric, optical and radio transmission.

In the Sixties there became evident the rise of distributed communication in automation with the emergence of the Distributed Control System-DCS. This system had the tendency to distribute its control functions, guaranteeing that if one control station failed, only part of the process would stop, leaving the rest of the process in operation.

The use of communication networks brought great advances in the automation of processes. However, a large obstacle to inter-process communication arose from the lack of standardisation of communication protocols.

From the rising of the OSI-ISO model, the makers of equipment started to use their communication Drivers as part of the application programme and only through them one could access the data stored in their equipment. This strategy made difficult the communication between the different makers of equipment. In order to eliminate the communication barriers created by the lack of standards between makers of equipment there was created, in 1996, the OPC-Foundation [1]. This organisation aimed to establish rules of communication between applications, encapsulating Drivers of manufacturers and offering standard communication interfaces, Figure 1. This standardisation procedure was called OLE for Process Control and its implementation is centralised in the Distributed Component Object Model - DCOM for objects distributed by Microsoft. The standard OPC has evolved through time, bringing Open standards and independence of platforms, as can be seen through the specification OPC XML. [2]

Not withstanding the standardisation offered by OPC, we can observe that the use of this technology is still tied to a given platform, rendering difficult the use by the new systems, thus limiting its usage to the former. For some time
researchers in the area have been considering alternatives for the integration of automation processes that would compensate for the deficiencies present in the current systems. The change of the paradigm of the technologies geared to objects to technologies oriented towards services is showing to be a promising alternative.

In this way we can conclude this section of our work saying that the ways of transmission within industrial automation is a combination of an extremely rich channel of transmission between the factory-floor unit and the control systems, and as such is the permanent object of evaluation and improvement.

III. WEB SERVICE

The integration between corporative applications has been the subject of constant research by researchers, scholars and industrialists. As a result, there are several new technologies geared to solve the problem related to the integration between corporative applications. Among them we can name CORBA, DCOM, JAVA, RMI among others [3]. Each technology has a set of rules and services of Middleware capable of implementing a nucleus of required functions, table I.

However, the arrival of Service Oriented Architecture-SOA restarted discussion of integration of corporative systems. In the last few years, companies have used SOA in the integration of corporative applications with good results [4]. In this context, the Web Service stimulated this type of architecture, even though they were initially viewed as one more embodiment of the technology of distributed objects [5].

The Web Services are a technology that is more and more used and that could occupy gaps left by the speed with which the existing technology evolves in the commercial sector as well as in industry where the technological parks are highly heterogeneous and are subject to frequent evolutionary changes [6].

For the working group of the World Wide Web - W3C, the definition of Web Service is: “A system of software devised to support interoperability between computers in net using HTTP rules with the serialisation of the XML documents” [7].

There are three basic technologies around Web Services

- Simple Object Access Protocol - SOAP
- Web Service Description Language - WSDL
- Universal Description, Discovery and Integration - UDDI

Figure 2 illustrates how to open the process of publishing and access to a Web Service by a client. The process can be summarised thus:

1) **Registration** This is the first step by which a client passes to have access to the functionings of the Web Service, or rather, the service provider should register its services with a UDDI register.

2) **Search** The client, then, searches for the service in the UDDI register.

3) **Description** The client obtains a description of the variable methods and types of parameters of the provider, this is done through WSDL, that identifies which services are available in the Web Service.

4) **Communication** The client, then, invokes the Web Service utilising SOAP, which functions with a grammar for the XML existing in the requests and responses between user and Web Service.

A. **SOAP**

SOAP is one of fundamental element of Web Services because it composes the base of the stack of communication.

<table>
<thead>
<tr>
<th>Table I Technologies Comparison Table.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Format</strong></td>
</tr>
<tr>
<td>DCE</td>
</tr>
<tr>
<td>CORBA</td>
</tr>
<tr>
<td>JAVA RMI</td>
</tr>
</tbody>
</table>
protocols. Unlike the protocols used by technologies such as CORBA, COM and DCOM, SOAP uses XML to codify the messages, by adding more strength and legibility because of the greater use of XML in applications, based particularly in the Web [8]. SOAP protocol is divided into three basic elements:

- **Envelope** - is the principal element of XML that represents the message.
- **Header** - is a generic mechanism of addition of characteristics to the SOAP message in a decentralised manner, without prior agreement between the communicating parts.
- **Body** - contains the actual codification of a call to a method and all the arguments of access or a codified answer that contains the result of a call to a method.

Because SOAP uses XML for its definition, any type of data could be represented, so long as it can be specified in one XML Scheme.

**B. WSDL**

WSDL is a language used to describe the services in XML that has the function to show the methods available, their locations and the rules that a Web Service offers. The idea is similar to a Java interface, or an Interface Definition Language-IDL CORBA. As such, WSDL offers independence of language and components independent of the type of technology used [9].

**C. UDDI**

UDDI is a mechanism that offers a unified systematic to search the service providers through a centralised register service. Making an analogy with the search for a Web site, the UDDI is equivalent to a search service like Google, Yahoo among others. Thus, the service provider register to the UDDI Registry and the users can localise the services through the register.

The UDDI Registers allow the user to locate business and services in three ways.

- “**Yellow Pages**” - Catalogue business based on business and type of services.
- “**White pages**” - Catalogue business by name and contact details.
- “**Green pages**” - Catalogue information on services offered by a given business.

In this way we can conclude this section by saying that the systems based on an architecture oriented to services have as characteristic a weak coupling and dynamic connections between their modules, unlike the paradigm oriented towards objects, that possess strong coupling and static connections between their modules. Um example of this type of architecture is the OPC seen in the previous section.

The possibility encapsulation of the Drivers of the manufacturer by a Web Service, as shown in Figure 3, will permit that clients access to information of the Clips aggregating the following advantages:

- Service mobility
- Automatic support to several users
- Greater reuse
- Greater upward mobility
- Greater availability

**IV. RESULTS**

Within the communication data models shown above, this work implements a new way of communication using Web
Service and open technologies with the objective to facilitate the Management of Information of the Industrial Process. This approach proposes the attainment of a differential from the methods used in the industries at the present, through a change of focus from the existing programmes oriented to objects, to a programme distributed oriented to the service. Figure 4 illustrates the basic components of the architecture.

![Fig. 4. Architecture overview.](image)

Because this work uses an implementation of Java of Web Services, it was necessary to use API Java Native Interface-JNI to make possible the loading and use of Drivers within the Web Service. The JNI is a native interface programme. It permits the native codes to be executed within the Machine Virtual Java to interoperate with applications and libraries written in other programme languages such as C, C++ and Assembly [10].

Figure 5 shows that for Web Service to use the Drivers of access to PLC it was necessary to implement an adapter (DLL - Adapter). This adapter is responsible for encapsulate the function (write-read) of access to PLC and make them available to the user through the Web Service. From this architecture on is possible to observe the flexibility of the clients, from the use of any operational system to the development of client applications, independent of the programme language.

![Fig. 5. Architecture detailed view.](image)

In the server machine (Web Service) were used the following artefacts of software:
- Java Development Kit 1.5, for the development of class of access to data and encapsulation of the driver of the PLC [10].
- Apache Tomcat 5.5 as server Web [11].
- Apache Axis to create Web Service [12].

A didactic plan was created, Figure 6, which had a water reservoir, whose level was Controlled by a PLC SIEMENS, type S7-200 that received a signal 4 to 20mA in one of its analogous entries. To control the level of water in the reservoir was used a Pressure Transmitter(LD).

![Fig. 6. Plant of simulation.](image)

To allow the clients to have access the data of the plan via PLC, a client application was developed that communicated with the server (Web Service) through a transport protocol HTTP, allowing the on-line visualisation and manipulation of existing data on the PLC. For the monitoring of the water reservoir was elaborated an interface that would better represent the didactic plan. See Figure 7.

![Fig. 7. Monitoring screen.](image)

From the instant the process (didactic plan) and the Web Service began, the clients started to receive data of the PLC. Both the clients and the Web Service were installed in a segment of the Ethernet network with around thirty machines gaining varied access to the network and even so the replies to the clients’ queries were around 200ms. At present, there exist several solutions for the acquisition of data from the factory floor. The OPC protocol is one of the better known and adopts the concept of encapsulating the Drivers of the makers of PLCs as this work suggests. However, the OPC adopts the use of distribution of programmes to objects and is based on the technology DCOM of the Microsoft. With the
utilisation of a service oriented architecture we can establish that we can offer the same services that technologies such as OPC offer, adding characteristics such as interpolability, portability, scalability among others.

V. CONCLUSION

The industrial automation sector has undergone widely-known evolution impelled mainly by Information Technology, which is refining this sequence more and more. Even with all this refinement, one finds strong resistance to the new standards, particularly the open standards, and the sector continues to rely on the closed standards making more difficult the adoption of new and modern technologies. The use of technologies based on the open standards has not been totally accepted and is not, yet, a reality within the industrial sector. This is the result of a conservative attitude adopted by this sector, which restricts the development of applications to proprietary technologies. The adoption of architecture oriented to services provides a series of advantages when one tries to access the factory-floor data and to disseminate it to other sectors of an organisation. The use of the Web Services brought this reality to the surface. The independence of access to these (Web Services) makes it possible that clients can have access to the diverse data of an automation process through Browser, Desktops applications or even mobile equipment like cellular phones and PDAs.

Thus, we may conclude that, in the near future, the architecture oriented to services will be effectively applied in industry. This actuation will be observed from the factory floor to corporate decision-making levels. Such a change reflects the actual needs of automation processes such as portability, interoperability, robustness and scalability.

REFERENCES