Design of a HTML5 SCADA System

Yamuna Maccarana*, Alberto Cologni*, Fabio Previdi*
*Università degli Studi di Bergamo,
alberto.cologni@unibg.it

***Abstract*—The SCADA (Supervisory Control And Data Acquisition) systems are one of the most important components of the industrial automation. They are used as the interfaces for production machines, lines and entire plants. The industrial solutions are usually based on desktop applications and require a continuous presence of operators close to the HMI (Human Machine Interface).**

In the last decade, the evolution of the internet and all the correlated technologies allowed to extend the ways of communicate, interact and manage information: in this context the evolution of the industrial world must engage the consumer world, introducing innovation in the processes. A HTML5 solution for the SCADA allows to extend the system capabilities as the possibility of distribute the HMI on different hardware architectures, activate push notifications of alarms, establish the direct connection of the plant to the ERP and the company management systems.

This paper presents the prototype of a HTML5 SCADA: this solution contributes to the development of a sophisticated context awareness system, allowing to filter and provide the needed information to specific users. The target is to achieve all this together with reactive interfaces to provide good performance (considering the web data transmission bottleneck).

I. Introduction

The strong innovation, started from the birth of the internet, moved the industry to a new industrial revolution: the industry 4.0, entered in the vocabulary and the European programs since 2010, describes fundamentally the extension of the IT (Information Technology) to the low level systems, defining the common term IoT (Internet of Things).

Most of the industries, such as refineries, chemical factories, electric power generation plants, and manufacturers have large, distributed and complexes plants. The earliest control networks were simple point-to-point networks connecting a monitoring or command device to a remote sensor or actuator. However, because of the necessity of the operators to continuously monitor and control more sections of the plant to ensure its proper operation, such systems have lead to the development of networking technology that has made feasible to command and control the plant remotely.

Companies that are members of standardization committees (e.g. OPC, OLE for Process Control) and are thus setting the trends in matters of IT technologies generally develop SCADA (Supervisory Control and Data Acquisition) systems [1]. Conventional SCADA systems use PC, notebook, thin client, and PDA as a client, but also, Java-enabled mobile phones have been achieved as clients, in order to display and supervise the position of a sample prototype crane via mobile phone [2].

This step allows many embedded devices and Programmable Logic Controllers (PLCs) to provide data directly via Transmission Control Protocol / Internet Protocol (TCP/IP) and Web services. Additionally to that, there is a trend from proprietary visualization software to the use of Web-based visualizations [3]. During the last decades, this remote command and control has been made feasible due to the development of networking technologies and the advent of Industrial Control Systems (ICS). ICS are command and control networks and systems designed to support industrial processes [4], [5], [6]. All this have since evolved into complex networks that support communication between a central control unit and multiple remote units on a common communication bus. The nodes on these networks are usually special purpose embedded computing devices such as sensors, actuators, and PLCs. The largest subgroup of ICS is composed of industrial command and control networks, that are commonly called SCADA networks.

In this context, the extension of the HMI (Human Machine Interface) to a cloud approach allowed the HMI split into server and visualization device, enabled the server device to connect with controller and proceeded protocol conversion, data logging, event logging, recipe, database maintenance, macro commands execution, etc. There are several technologies for Web Interfaces and the communication between client and server. ASP.NET SignalR, to name some, is a library for ASP.NET developers that simplifies the process of adding real-time web functionality to applications. Real-time web functionality is the ability to have server code push content to connected clients instantly as it becomes available, rather than having the server wait for a client to request new data. ASP.NET Web API is a framework for building web APIs on top of the .NET Framework. WFC is the unified Microsoft programming model for the implementation of service oriented applications. Nevertheless, some technological barriers make this solution not applicable yet. Most of the interfaces provided by the devices do not match the requirements of modern Web technology required for Web-based visualizations. This leaves a gap between the device and the visualization and introduces additional transformation of the data and complexity. OPC-UA (OPC Unified Architecture) [8] could solve this problem, thus, we aim to fill the gap between all these requirements, introducing and integrating new and already existing technologies. Advantages will result in terms of reduced development time and costs, predisposition to a cloud solution, management outlay reduction, possibility to access to the plant remotely, guarantee of multiplatform operability and openness to big-

978-1-5090-1131-5/16/$31.00 ©2016 IEEE

data solutions.

To do so, we propose a SCADA system that is based on layers, as depicted in Fig. 1. In particular, Application Layers are used to create sessions for users and products authentication and authorization. To create secure channels for the communication, Communication Layers are introduced to allow applications authentication and authorization, guaranteeing integrity and confidentiality. In the end, the socket connection is managed by the Transport Layers, which provide the availability of the entire communication system. In the last years, a lot of application has been moved to the cloud, and based on web interfaces. The first step in this direction is to implement a web solution based on a local web server (nowadays the internet connection is not always fast enough to support a time critical data stream). In literature it is possible to fine some HTML SCADA applications [7]. Moreover, in literature, there is not trace of a solution implemented by the use of the HTML5 markup language. Because we also want to improve the support for the latest multimedia, we will achieve all this programming in HTML5.

This paper is organized as follows. In section II, the system structure is proposed, with description of services enabled by our solution. In section III, the implementation of our system is presented in details, highlighting the brought advantages. Then, in section IV, a case study is described, with references to the obtained results. Section V shows a final overview of the system and suggests future directions.

II. System structure

Our aim is to achieve a web graphic interface that allows to visualize and monitor plant data, to set machine parameters and to manage alarms. We aim to achieve all this by reading and writing data in the database through communication logics between the plant, the database and the web server.

For better understanding, a depiction of the proposed system architecture is reported in Fig. 2. Data should be managed, acquired and provided from the plant to the user passing through the following components:

- a SCADA server, composed of a communication manager, a web server and a database;
- Internet, in a cloud-like configuration;
- personal devices, tablet and PCs.

This architecture also allows the introduction of new services such as remote GUI (Graphical User Interface), event-based push notifications, supply of information related to production statistical surveys and much more. It comes natural that all this is meant to improve several factors as productivity, easier and quicker information access, critical event constant monitoring and advice, etc.

Indeed, the improvements in the visual interface of the SCADA GUI provided fairly high resolution graphical animations [9], useful to guarantee for the user the quickest and clearest access to all (or part of) the information acquired remotely on the plant. Because the classic approach to plants was sinuous and complex, users who are not expert may have been discouraged by the difficulty of accessing data in such

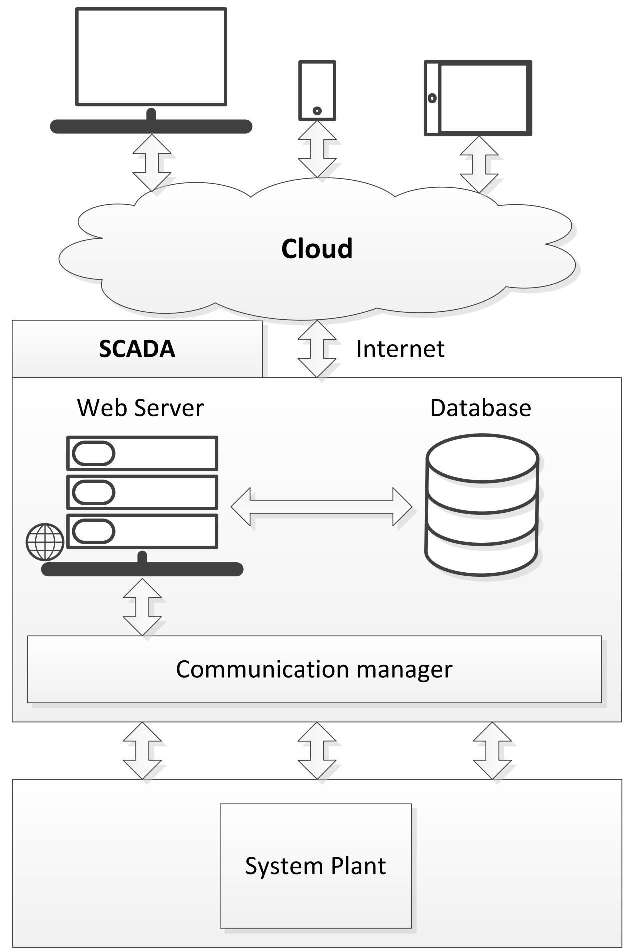

Fig. 2. System Architecture

way. The SCADA-User Interface is programmed with multiple user access levels, so that data are provided to the user with the minimum possible effort. Such access is protected to ensure the authorized user to view the status of the remote system and provide further assistance to the plant engineers.

Moreover, event-based push notifications on mobile personal devices remain the perfect solution to provide not only the necessary information to the the operator, but also to provide them only when needed. This way, the operator will be free to take no interest in the system except of when strictly necessary.

III. Implementation

In order to move toward the seamless standardisation and integration of SCADA systems, some technological solutions have been adopted. To start with, we relied on the OPC foundation, that defined a series of software interfaces for the standardisation of the information flow from process level to management level [10].

To the standard OPC based on Microsoft COM/DCOM, which has been deemphasized by Microsoft itself [11], we preferred to adopt the OPC-UA, which brings along the

978-1-5090-1131-5/16/$31.00 ©2016 IEEE

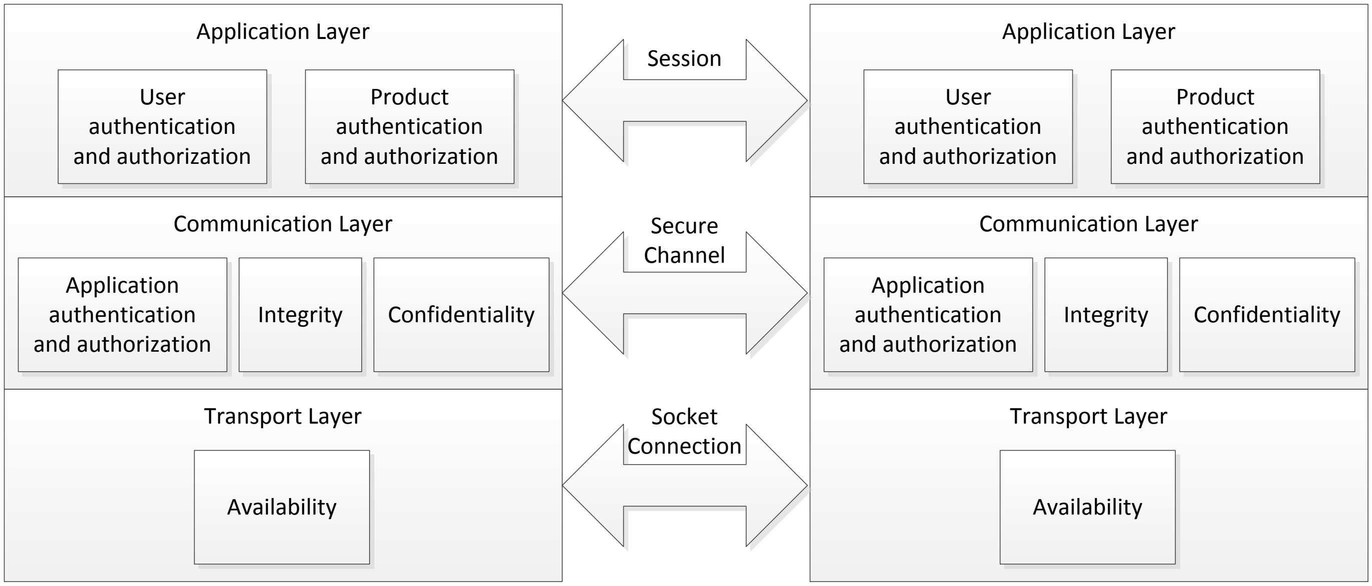

Fig. 1. System layers

advantages of using the OPC XML-DA (e.g. it is not provided by any software platform) and overtakes its limitation related to the lack of performance introduced by the use of XML. The OPC-UA, even if used as a client-server architecture, typically is an application that holds both client and server role. This happens because, in the device to device communication, the server side is integrated as well as the client one. A typical OPC-UA application, thus, is composed by three different software layers, as depicted in Fig. 3.

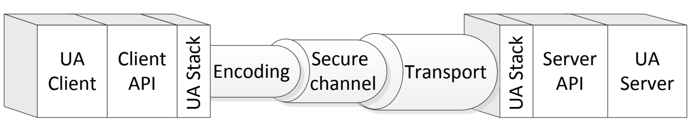

Fig. 3. OPC-UA layers

The first one is the UA Server or Client, connected to the UA Stack by either Server or Client APIs. Then the channel comprise a phase of encoding and transport over a secure channel.

The services offered by the standard OPC-UA are:

- discovery service;
- session service;
- subscription service;
- read and write service;
- view service;
- secure channel service;
- query service;
- method service.

All these services are meant to facilitate a quick and secure access to data.

In addition to all this, the publish-subscriber communication lays the foundation to enable push notifications. As said in Section II, push notifications are a key feature of the proposed system, in order to allow maximum workload relief to the user.

Moreover, thanks to its architecture, this solution is scalable on the number of the devices in the system. This helps toward the optimisation of resources in each configuration, from big industries to small plants with few users.

Another important characteristic of our system, is that all this is achievable for free for most of the PLC platforms.

On OPC-UA, an application relying on .NET 4 framework server-side web application has been created using the C# programming language. This technological choice allowed to facilitate its maintainability integration with most of the databases thanks to the high level of the used framework.

To support all this, an IIS Server Web, including Internet Information Services 7, a unified Web platform that integrates IIS, ASP.NET, Windows Communication Foundation and Windows SharePoint Services.

Following the theory furnished in [12], our system prototype is composed by several modules, as depicted in Fig. 4.

The first component is a Client component such as a web browser to be used for the execution and visualization of web pages of the projected SCADA system. Moreover, it is able to send requests to update web pages with the new data. Requests will be achieved asynchronously in the background by the AJAX Engine, so that the web pages results reactive to the user. When the server response gets to the AJAX Engine, the latter forwards the data from the web page. Then, a new request is send only if the previous requests has ended well. Also, another bidirectional channel can be used to write reference values directly from the control process.

The second component is a Web Server that awaits the client requests and forwards them to the Web Service. The latter executes the appropriate web method:

978-1-5090-1131-5/16/$31.00 ©2016 IEEE

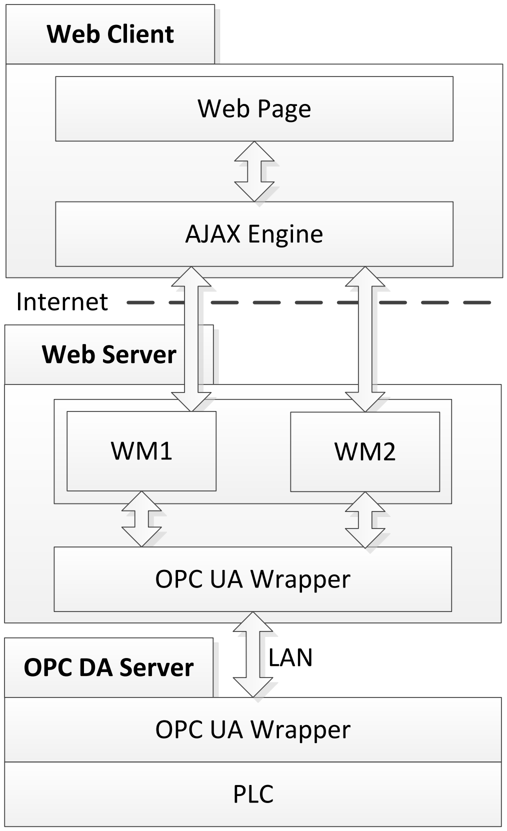

Fig. 4. System architecture prototype

- WM1 if the requests is about new data;
- WM2 if the requests contain references to the OPC server.

The WM1 server waits for a data update in the cache of the OPC DA server and then provides the new data to the client. The WM2 server writes the following setpoint of the client to the OPC UA server without waiting.

The third component is the OPC UA server, which is the interface to access the physical control of the process (PLC). This server has a layer that provides the standard http interface to its client.

The last component is the PLC, that contains the logic to manage the control process itself.

IV. Case Study

The whole described system has been tested in a case study, in which a B&R PLC has been used and remotely monitored through the web SCADA application. The technologies that have been employed in the project are reported next:

- Web Server -IIS Express;
- Server OPCUA for B&R;
- SDK unified automation;
- AJAX and Web Services;
- C#.NET 2012 programming language (server side scripting);
- Javascript programming language (client side scripting);
- Microsoft developer Visual Web 2012 (for active web pages design);
- PC with Windows 7 OS.

In this application a PLC B&R X20CP1586 has been employed: its main advantage is that, in such devices, the server OPCUA is already integrated, and thus there is no need to implement an OPCUA server to interface with the PLC. This allows to reduce developing time of the Web application.

The supported profiles are reported next:

- Security Policy None;
- Core Server Facet;
- Low End Embedded Server Facet;
- DataAccess Server Facet;
- Basic DataChange Subscription Server Facet;
- Enhanced DataChange Subscription Server Facet;
- Event Subscription Server Facet;
- Method Server Facet.

A web page is shown as example in Fig. 5.

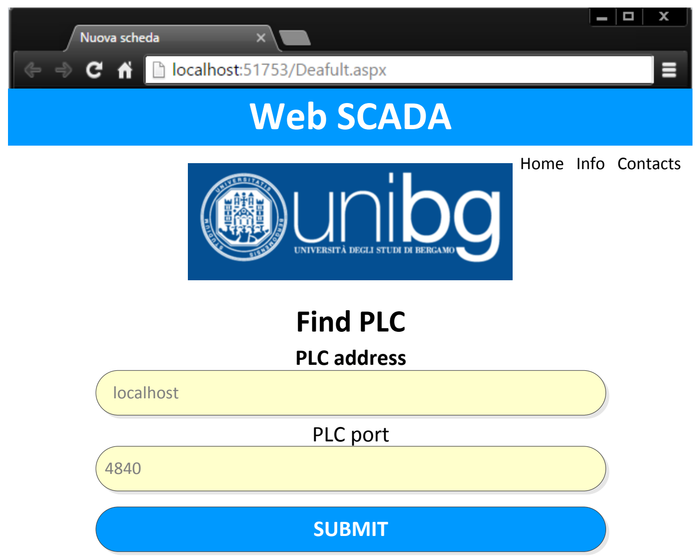

Fig. 5. Web page example

The use of HTML5 results evident to the user in terms of graphical impact and it plays an important role in terms of enhancing the graphical visualization of specific data. Such solution is meant to work on most of the PC, tablet and personal devices.

The advantages of using a B&R PLC relies in:

- its versatile programming in IEC 61131-3, C, C++ and CFC languages;
- its scalable hardware platform with long term availability;
- its open standards and bidirectional interfaces;
- its easy configuration;
- its OPC-UA server integration.

However, the advantages of using such a PLC bring with them one cons: its cost. Nevertheless, the results justify such cost and prove the feasibility of the proposed prototype.

The case study represents a demo application that allows to manage an axis controlled by a brushless motor. The demo consist of a series of web pages that allow to:

- Connect to a OPC-UA server
- Turn on and off the axis
- Define a speed reference for the motor
- Manage the axis alarms (subscribe and unsubscribe the axis)

This demonstrator allows also to activate and deactivate the log of the signals and the operations to the database.

This demo represents the core of a SCADA application: all the most important feature are demonstrated, and the key point of the Server - PLC connection have been tested.

V. Conclusions

In this paper the seamless integration of a Web SCADA system application with existing industrial plants and machinery has been discussed. The document illustrated a new SCADA system based on layers comprising technologies such as OPC-UA and .NET and relying on a MySQL standard database. The presented framework has been implemented in C#, which is the perfect candidate for programming on different development environment and absolutely aligned with all the system used with the actual SCADA solutions. This happens also thanks to the .NET software development framework from Microsoft that supports many languages providing all the necessary libraries. A strong innovation to this solution has been given by the use of HTML5.

The results demonstrate that this solution provides advantages in terms of efficiency, cost and constant remote monitoring of the system and supports specific and dedicated information access to each user. All the previously mentioned advantages have evidently shown from the very first use test.

Although, there exist limitations regarding OPC-UA servers included starting from Automation Runtime version B4.04, as shown next:

- 100 maximum subscription for client connection;
- 1000 maximum monitored items for subscription;
- 100000 maximum registered nodes for namespace;
- 10ms minimum sampling time.

As future work, we aim at implementing and extending the proposed notification services, introducing more and more improved features, which will be customizable based on context. The main target is to enhance the system with attention to the actual hardware specifications, together with a strong focus on protocols extension.

However, the future system is thought to be a configurable system, also with a graphical tool to personalize the architecture and allow easy GUI management and composition. The direction is the one of allowing the user to manage all at higher and higher level, with consequent lower time effort and costs.

References

[1] Daneels, Axel, and Wayne Salter, *What is SCADA?*, International Conference on Accelerator and Large Experimental Physics Control Systems, Trieste 1999.

[2] Engin Ozdemir, Mevlut Karacor, *Mobile phone based SCADA for industrial automation*, ISA Transactions, Volume 45, Number 1, pages 6775, 2006.

[3] Patrick Roland Gansterer, *Web Process Control Protocol*, A WebSocket subprotocol for automation, 2015.

[4] European Union Agency for Network and Information Security (ENISA), *Industrial Control Systems/SCADA*, Protecting Industrial Control Systems, Deliverable, 2011.

[5] Simone Formentin, Alberto L. Cologni, Fabio Previdi, Sergio M. Savaresi, *A Data-Driven Approach to Control of Batch Processes With an Application to a Gravimetric Blender*, IEEE Transactions on Industrial Electronics, Volume 61, Issue 11, pages 6383-6390, November 2014.

[6] Michele Ermidoro, Simone Formentin, Alberto L. Cologni, Fabio Previdi, Sergio M Savaresi, *On time-optimal anti-sway controller design for bridge cranes*, American Control Conference (ACC), pages 2809-2814, 2014.

[7] A. M. Alexandru, Alice De Mauro, Maurizio Fiaschè, Francesco G. Sisca, Marco Taisch, Luca Fasanotti, Piergiorgio Grasseni, *A smart web-based maintenance system for a smart manufacturing environment*, Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2015 IEEE 1st International Forum on, pages 579-584, 2015.

[8] *ISO/IEC. IEC 62541: OPC Unified Architecture*, 2008.

[9] Nikita Shingre, Reema Nagwekar, Rupa Roy, Trupti Shendkar, Rupinder Kaur, *Reqwiev on Mobile Phone Based SCADA for Industrial Automation*, International Journal of Technical Research and Applications e-ISSN: 2320-8163, pages 1-4, March 2016.

[10] Al Chisholm, *OPC/OLE for Process Control Overview*, presented at the World Batch Forum, 1998.

[11] Stefan-Helmut Leitner, Wolfgang Mahnke, *OPC UA Service-oriented Architecture for Industrial Applications*, ABB Corporate Research Center, 2006.

[12] Hosny Ahmed Abbas Ahmed, *Efficient Web-Based SCADA System*, Automation Engineer in Qena Paper Company, June 2011.