PERFORMANCE EVALUATION OF A SCADA SYSTEM  
FOR MONITORING AND CONTROLLING THE  
INDUSTRIAL PROCESSES  

BY  
SIMONA-ANDA GHERASIM* and VASILE-GHEORGHIȚĂ GĂITAN  

“Ștefan cel Mare” University of Suceava,  
Faculty of Electrical Engineering and Computer Science  

Received: February 18, 2013  
Accepted for publication: March 20, 2013  

Abstract. In recent decades, SCADA systems had an exponential trend of development, being used all over the world, particularly to monitor and control industrial processes or infrastructure. This article presents a system for monitoring and control of industrial processes that is part of a complex SCADA system. Based on this system, we proposed a solution that validates the possibility of extending the functionality of MCPI client, by porting it on a real-time operating system, Windows Embedded Compact (CE). Application testing and analysis of results have validated the proposed solution.  

Key words: middleware, SCADA systems, embedded systems, data stream, performance.  

2010 Mathematics Subject Classification: 68M99, 68W35.  

1. Introduction  

Knowing a rapid development in recent years, today's SCADA systems are found everywhere, thus highlighting a variety of architectures. The main advantages of such systems are: increased efficiency, lower costs and increased profits.  

*Corresponding author; e-mail: sgherasim@eed.usv.ro
SCADA systems allow users to collect data from one or more processes within a certain distance, and control them.

If a process is spread over a fairly large area (tens or hundreds of kilometers), such as a pipeline distribution system, the benefits of a SCADA system can be appreciated much better, thus reducing costs visits which routinely monitor the operation of plant/processes.

In other words, a SCADA system allows an operator located in a central location of processes distributed over a wide area, to carry out different processes change references to remote controls, monitor alarms, open or close switches or valves to gather information measured from certain processes, etc.

The main functions that must satisfy a SCADA application are:
- data acquisition,
- data communication network,
- presentation of data and
- control system.

The four functions are performed only by means of components, such as sensors, control relays, Remote Terminal Units - RTU (Remote Terminal Units), a master station and communications infrastructure (Arghira et al., 2011).

2. Related Work

One of the most difficult requirements that must achieve a middleware application is data exchange.

Starting from applications based on COM/DCOM, the most important specification for data exchange automation industry, were provided by the OPC Foundation. Thus, first specifications defined by OPC Foundation are: OPC Data Access (DA), OPC Alarm & Events (A & E), OPC Historical Data Access (HDA), then OPC Express Interface (OPC now named .NET) (Sahin & Bolat, 2009). They are widely accepted in the industry and are already implemented by any system of automated industry.

OPC middleware products are designed to enhance any application based on OPC technology. OPC specification simplifies data handling in client-server applications, significantly improve transit data from OPC interfaces and ensure reliability in communication.

Initially, the OPC has been defined as a standard solution for recurring tasks for connecting different applications (HMI or SCADA) with automation and process control (Cavalieri et al., 2010). Today, OPC technology provides interoperability between: two devices, a device and a software application, two different software applications, a server and a client OPC UA, OPC UA, OPC UA server and OPC client classic a classic OPC server OPC UA server.

OPC specifications help simplify integration HMI or SCADA type applications (Zhang & Feng, 2010; Hongli & Feng, 2011).
3. SCADA System Functionality

Based on heterogeneous distributed real-time system oriented on SCADA applications presented in (Găitan et al., 2010), has developed a system for monitoring and control of industrial processes. The architecture of this system is shown in Fig. 1.

![Diagram of SCADA System Functionality](image)

Fig. 1 – Architecture for real-time heterogeneous distributed systems oriented SCADA applications (Găitan et al., 2010).
The system is built on the following software modules:

- **Client application**
- **Middleware bus**
- **Server application**

The client application communicates with the server application via a middleware bus (Găitan *et al.*, 2008).

The application server has the following components:

- **Data Servers**
- **Communication software component**
- **Data acquisition modules represented by device drivers**

When a client starts a request to the application server, and requests access to a particular data, the data server will retrieve the required data from the communication component.

The flow of data is retrieved from the network devices by the acquisition module and stored in the object dictionary from the communication component. The data dictionary will contain a set of objects and data members for each network device, taken from the EDS file attached.

The devices respond to different sets of commands, but the data acquired from these devices do not have a standard format. Therefore, the acquisition component must implement one driver for each device.

The driver’s task is to map the device-specific commands and responses to be understood by the data acquisition component.

### 4. Testing the Application

To test the client application developed for the real-time operating system Windows Embedded CE presented in (Gherasim *et al.*, 2012), I used a star-type architecture, composed of 6 eBox 2300 SX embedded systems, 1 PDX 089T embedded system, a desktop PC and a Super Stack II Baseline 10/100 Switch. Testing the network architecture of the application is shown in Fig. 2.

The flow of data sent to the local network was captured with Colasoft Capsa Enterprise software tool installed on the desktop computer. Colasoft Capsa Enterprise allows intercepting and analyzing Ethernet frames in terms of transmission speed.

On the 7 embedded systems were launched running one data server gpDAServer and a client MCPI ported to Windows CE. In this way, each client application has connected to the server via an OPC object, able to read or write data. The OPC object enables the refresh rate setting for data written or read from the server.
The server interface is shown in Fig. 3.
Desktop computer was used to monitor any changes in servers that were running on the 7 eBox 2300 SX embedded systems. I used a MCPI client running on a Windows XP operating system, with which I am connected to the 7 servers to be able to see data updates.

Tests were conducted using different refresh rates for data transmission to the server. Minimum refresh rate was 100 ms (0.1 sec) and the maximum used in the tests was 1 sec. Also, each item was tested OPC data transfer speed for 1 item, 5 items, 10 items and 15 items.

In Figs. 4 to 7, there are four screenshots with the specified number of items (1 item, 5 items, 10 items and 15 items) for each of the 7 items connected to the OPC server.

![Fig. 4 – Testing the application with 1 item for each of the 7 OPC objects connected to server.](image)

![Fig. 5 – Testing the application with 5 items for each of the 7 OPC objects connected to server.](image)
Fig. 6 – Testing the application with 10 items for each of the 7 OPC objects connected to server.

Fig. 7 – Testing the application with 15 items for each of the 7 OPC objects connected to server.
Table 1 shows the experimental results obtained from tests at different update rates to different numbers of items in a group.

<table>
<thead>
<tr>
<th>Number of items</th>
<th>Refresh rate (Kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100 ms</td>
</tr>
<tr>
<td>1 item</td>
<td>191.936</td>
</tr>
<tr>
<td>5 items</td>
<td>205.984</td>
</tr>
<tr>
<td>10 items</td>
<td>343.568</td>
</tr>
<tr>
<td>15 items</td>
<td>233.12</td>
</tr>
</tbody>
</table>

Fig. 8 – Results diagram.

Fig. 9 – 3D results diagram.
Similarly, the testing validation for the client application developed for Windows Embedded CE, the same tests were conducted on a similar architecture.

This time, a star-type architecture consisted of 8 PCs with XP operating systems connected to a TRENDnet switch type TEG224WS+, with a 100Mbps Ethernet interface.

The 7 eBox systems (PDX 089T system included) of previous test architecture were replaced with 7 computers, and remained Master Station with the same role.

On the 7 stations were launched running one database server and a client gpDAServer MCPI running on Windows XP. In this way, each client application has connected to the server via an OPC object, able to read or write data. OPC Items for determining the discount rate for data written or read from the server.

Table 2 shows the experimental results obtained from testing traditional client application on the test network architecture shown in Fig. 9, using the same discount rate and the same number of items in a group as in the first test.
Table 2

<table>
<thead>
<tr>
<th>Number of items</th>
<th>Refresh rate 100 ms</th>
<th>500 ms</th>
<th>1 sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 item</td>
<td>208.32</td>
<td>50.272</td>
<td>24.416</td>
</tr>
<tr>
<td>5 items</td>
<td>315.84</td>
<td>70.848</td>
<td>35.168</td>
</tr>
<tr>
<td>10 items</td>
<td>441.28</td>
<td>86.976</td>
<td>42.592</td>
</tr>
<tr>
<td>15 items</td>
<td>528.32</td>
<td>89.792</td>
<td>44.512</td>
</tr>
</tbody>
</table>

Fig. 11 – Results diagram.

Fig. 12 – 3D results diagram.
5. Conclusions

From the analysis results can be drawn the following conclusions:

1. In both cases, there is a downward trend in the data transfer speed with decreasing discount rate of 100 ms. to 1 sec.

2. If there are small differences in data transfer rates between the two types of operating systems, these differences are due to the possibility that the server do not refresh rate to time because of overloaded processor.

3. Another explanation could be the way of implementing TCP/IP on Windows Embedded CE.

4. The higher the discount rate, data transfer speed values for testing Windows CE approaching values obtained on Windows XP.

Acknowledgments. This work was supported by the project “Knowledge provocation and development through doctoral research PRODOCT 1 Contract no. POSDRU/88/1.5/S/52946”, project cofounded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

REFERENCES


EVALUAREA PERFORMANȚELOR UNUI SISTEM SCADA DE MONITORIZARE ȘI CONTROL AL PROCESELOR INDUSTRIALE

(Rezumat)

În ultimele decenii, sistemele SCADA au avut o tendință de dezvoltare exponențială, fiind folosite peste tot în lume, în special pentru a monitoriza și controla diverse procese industriale sau de infrastructură. Principalele avantaje (pe termen lung) aduse de dezvoltarea acestor sisteme sunt: creșterea eficienței, scăderea costurilor și creșterea profitului.

În acest articol, este prezentat un sistem de monitorizare și control al proceselor industriale ce face parte dintr-un sistem complex de tip SCADA. Pe baza acestui sistem, am propus o soluție ce validează posibilitatea de extindere a funcționalității clientului MCPI prin portarea acestuia pe sistemul de operare în timp real, Windows Compact Embedded (CE).

În urma evaluării performanțelor sistemului prezentat și a analizei rezultatelor obținute se pot trage următoarele concluzii:

1. În ambele cazuri se observă o tendință de scădere a vitezei de transfer a datelor odată cu scăderea ratei de actualizare de la 100 ms la 1 sec.

2. Acolo unde există diferențe mici între vitezele de transfer ale datelor pe cele două tipuri de sisteme de operare, aceste diferențe se datorează posibilității ca serverul să nu facă rata de refresh la timp, din cauza procesorului prea încărcat.

3. O altă explicație ar putea fi modul implementare al stivei TCP/IP pe Windows Embedded CE.

4. La rate mai mici de actualizare, valorile vitezei de transfer a datelor în cazul testării pe Windows CE se apropie de valorile obținute pe Windows XP.

Testarea aplicației, precum și analiza rezultatelor au validat soluția propusă.