Internet-Based Production Monitoring and Reporting

Krzysztof Sacha and Wojciech Pikulski

Warsaw University of Technology, Warszawa, Poland
k.sacha@ia.pw.edu.pl, w.pikulski@elka.pw.edu.pl

Abstract. This paper describes a new approach to production data monitoring and reporting. The monitoring and reporting system called SMARP is composed of a small transponder, located on the plant floor, and a server, which can be located anywhere in the Internet. The main goal of SMARP is to provide the manufacturing decision maker with aggregated on-line process data in order to describe the effects of the plant operation, the effectiveness of the plant equipment and the causes of loses, such as accidents, damages and stoppages. The user of SMARP can also be the plant owner or any other authorized person, who can connect to the server through an arbitrary communication device, e.g. a laptop, a tablet or a mobile phone.

Keywords: production monitoring, SCADA, Internet based SCADA, manufacturing execution system.

1 Introduction

A typical plant consists of machines and devices arranged into production lines, and operating under the control of local controllers, which acquire data from the sensors and convey commands to the actuators of those machines and devices. Local controllers communicate through a computer network (a field bus) with supervisory controllers, which monitor and coordinate the plant operation. The main tasks of a Supervisory Control and Data Acquisition (SCADA) system are: Acquiring process data from the plant sensors, detecting alarms and abnormal situations, presenting the data to human operators and executing the operator commands. The scope of data and the way of presentation match the needs of an operator who controls the plant operation.

Control systems can be interfaced to an Enterprise Resource Planning (ERP) system, which looks at production orders and aggregates the process data to describe economic effects of the plant operation. SCADA server can be used as a gateway between the company's control and enterprise networks. The scope of data acquired and reported by ERP system match the needs of the manufacturing decision maker.

Typically, control systems reside on the plant floor and support on-line activities of the process operators, while the enterprise management systems reside on the main servers and support activities of the economic department of the company. The drawbacks are: High cost of the control and management systems, a dependence on particular vendors and restricted access to the process data, which is locked in a control room and accessible only to engineers trained to operate the proprietary systems.

The rest of the paper is organized as follows. Section 2 introduces SMARP system. Related work is discussed in Section 3. The architecture of SMARP server is shown in Section 4. An XML-based method for specifying the server is described in Section 5. A model of the server operation is introduced in Section 6. A discussion of the project status and the plans for future work are given in Conclusions.

2 Overview of SMARP

The described traditional SCADA-ERP architecture does not fit well the needs of the owners of small enterprises, who cannot afford such expensive systems, and who need a cheap solution with a possibility to monitor and control their businesses from a remote area. An alternative solution, much cheaper and more flexible than the traditional one, can be an Internet-based system for production monitoring, analysis and reporting (SMARP), which utilizes the public Internet infrastructure and offers a possibility to monitor and control the business aspects of the plant operation from anywhere in the world. The architecture of SMARP is shown in Fig. 1.

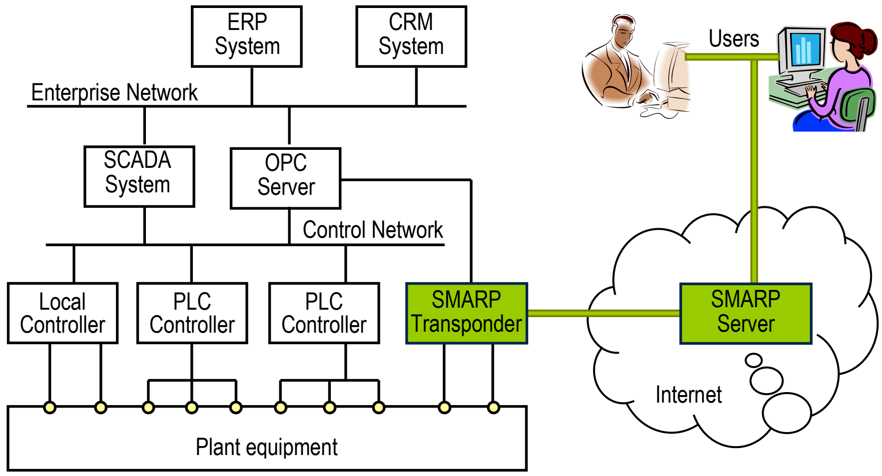

Fig. 1. The architecture of SMARP

The system consists of two basic components:

- SMARP transponder, which acquires the selected process data items;
- SMARP server, which stores, processes and presents the data to the users.

The transponder is a local device, which does not interfere with the plant operation. It acquires the process data from the controllers and additional sensors, and sends the data to the server through the Internet. The server is a computer which can be located anywhere in the network, e.g. on-site or in a cloud. The user of the server is the manufacturing decision maker, e.g., the plant owner, who can connect to the server through an arbitrary communication device, e.g. a laptop, a tablet or a mobile phone.

The advantages of SMARP architecture are the following:

- Low cost and ease of the system deployment.
- No need for individual programming of the server.
- High flexibility and the ability work in a cloud environment.
- Openness of the architecture and the communication standards.

3 Related Work

SCADA system provides the means to monitor and control plant devices from a central location. A typical system is deployed with proprietary software and hardware and with dedicated communication infrastructure. The technology is old and a general description can be found in many places, e.g. [1,2].

OPC is a technology developed by OPC Foundation in order to interface industrial controllers with popular PC computers. The first version (1998) was tied to Microsoft OLE and DCOM technologies, however, the latest version OPC UA (2012) is technology independent. A suit of standard specifications can be found in [3].

Internet SCADA (iSCADA) is a newer approach, characterized by the use of the Internet infrastructure to combine traditional SCADA design with the open communication protocols, services and data formats in order to deliver cost-effective SCADA solutions. The approach is still subject to research effort. One example is the system architecture described in [4,5], which consists of SCADA server and web server located at intranet, a firewall that separates intranet from the Internet, and a set of web clients. SCADA server supervises PLC controllers working over a control network and communicates with web server over the intranet. The clients can be located both: In intranet or in the Internet. They are computers, mobile phones or tablets, which use a standard web browser to access the web server. iSCADA solutions are offered by several vendors [6,7].

Our work fits partially into iSCADA approach; however, our goals are different. We do not want to build another SCADA system, to support process operators and to replace the existing systems. Instead, we intend to develop a production monitoring and reporting tool, to support the manufacture decision maker in controlling the economic effectiveness of the plant. Business scenarios can be the following.

The owner of a small enterprise that maintains a production line and a few auxiliary machines controlled by local controllers, neither needs, nor can afford the expensive SCADA system. Instead, he or she can deploy a simple transponder with few sensors, like photocells to register the products, and send the process data to a server located in the Internet cloud. The plant owner can rent an access service on subscription basis and pay small monthly installments, instead of paying for the entire investment. The service allows him to access the data by means of a web browser from an arbitrary place.

A huge enterprise with several production lines and full-fledged control and managements systems can maintain a proprietary SMARP server, installed on the plant floor. The system can serve management staff to analyze the overall equipment effectiveness (OEE), track and trace the products, and monitor the alarm conditions.

4 Architecture of SMARP Server

The conceptual architecture of SMARP server is shown in Fig. 2. The server receives a continuous stream of messages that convey process data sent by transponder (or transponders) attached to the plant installation. Messages can also be received from a touch panel, which can be attached to the transponder. An independent source of the server activity is a timer module, which counts time signals and triggers periodical computations within the server. All the input messages are temporarily stored in an input queue, from which they are fetched and processed in a sequential way.

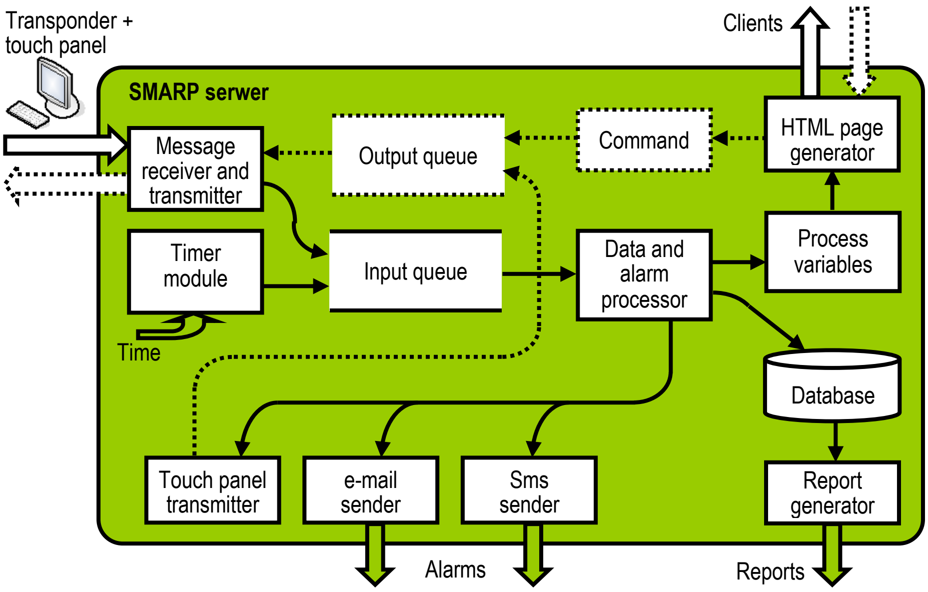

Fig. 2. The architecture of SMARP server

A single server can receive process data from several transponders, attached to the same or to different plants. Data related to the same plant are stored and processed jointly. Data related to different plants are treated separately. Thus, we can imagine that each plant has a separate instance of the server in our conceptual model.

The processing of data within the server consists of the following activities:

- Storage of the current values of measured variables received from the transponder.
- Computation and storage of the current values of derived variables, which have been calculated by the server using the values of measured variables.
- Detecting and handling of alarm conditions on variables.
- Storage of historical data (trend logs).
- Presentation of the plant operation using measured as well as derived variables.
- Notification of users about alarm conditions, by means of local touch panel, e-mail messages or SMS messages.

5 Specification of SMARP Server

All types of the process data stored within the server and all types of data processing are defined in a system specification, which is written in the form of a set of XML documents. The specification is compiled by an automatic tool and deployed on the server. No custom programming is needed in order to put a server (a server instance) into the operation. This enables fast deployment and helps in keeping low design cost.

SMARP specification consists of the following documents:

- types.xml – reusable library of types of measured and calculated variables;
- expressions.xml – reusable library of expressions to calculate derived variables;
- sensors.xml – definitions of measured variables;
- derived.xml – definitions of derived variables;
- constants.xml – definitions of constants, such as alarm thresholds;
- alarms.xml – definitions of alarms;
- factory.xml – description of the plant structure.

Types give semantic context to variables. A type has a name and a description of the data format, the physical entity, and the unit of measure. For example:

```
<types>
   <type name = "Temp"                type name
         data = "float"               data type, e.g. float, string etc.
         kind = "temperature"         physical entity
         unit = "C"                   unit of measure, e.g. Celsius
   />
   .......................            other type definitions
</types>
```

Sensors are process variables measured by sensors attached to the plant devices and transmitted by the transponder. A variable has a name, a set of attributes and a set of values. Names and attributes are specified in the following way:

```
<sensors>
   <item name  = "t1"                 variable name
         type  = "Temp"               type name
         mode  = "0"                  processing mode
         cycle = "10"                 measurement and reporting period
   </item>
   .......................            other variable definitions

</sensors>
```

Values of a measured as well as derived variables consist of the current value, time stamp and the current processing mode. Values are not specified in a file, however, they are stored and processed in memory during the server operation. The processing modes can be the following:

- 0 – variable is not processed;
- 1 – the value is stored in memory;
- 2 – the value is stored in memory and archived in the persistent history log.

Constants are defined as measured variables with processing mode set to 0.

Expressions define rules for calculating values of derived variables. An expression has a name, a list of formal parameters and a body written in XEXPR language developed by W3C [8]. Expressions can contain functions. For example:

```
<expressions>
   <expression name = "exp1" >    expression name
      <arguments>
         <item name = "x"         the first formal parameter
               type = "Temp"
         />
         ..................       other formal parameters
      </arguments>
      <expr>
         <add> <x/> <x/> </add>   XEXPR body (here: 2x)
      </expr>
   </expression>
   .....................          other expression definitions
</expressions>
```

Derived variables store the process data, which are not measurable directly, but which can be calculated from other variables (measurements) by means of expressions. The specification of derived variables is compatible with the specification of measured variables, but it contains a few additional attributes. For example:

```
<derived>
   <item name   = "l1"              variable name
         type   = "Unit_1"          variable type
         mode  = "0"                initial processing mode
         source = "exp2"            expression name
         order  = "120" >           sequence of computation
      <arguments>
         <item name = "t1" />       the first actual argument
         <item name = "a"  />       the second actual argument
         ..................         other actual arguments
      </arguments>
   </item>
   .......................          other variable definitions
</derived>
```

Alarms are abnormal situations, which may require intervention. Each alarm is associated with a derived variable called alarm reporting variable. If an alarm is raised, the variable is set to a positive value. Thus, an alarm is detected in the body of an expression used to calculate value of reporting variable. Alarm has a name, a report variable, and a list of panel, sms and e-mail addresses. For example:

```
<alarms>
   <alarm name   = "A_t1"                 alarm name
          report = "state" >              name of the report variable
      <phones>
         <sms>
            <to> 123456789 </to>          telephone number
            <note> text </note>           sms contents
         </sms>
         ..................               other sms definitions
      </phones>
      <mails>
         <mail>
            <to> xx@ex.com </to>          e-mail address
            <note> text </note>           e-mail contents
         </mail>
         ..................               other e-mail definitions
      </mails>
   </alarm>
   .......................                other alarm definitions
</alarms>
```

The structure of the plant equipment consists of machines, which are assembled into production lines. The structure and interconnections of those lines form a graph, which nodes are machines and edges are machine connections. Therefore, the plant structure is specified as a graph of machines, described in GraphML language [9].

6 A model of the server operation

SMARP server specification can automatically be compiled into tables stored in a relational database (Fig. 3).

Attributes of all the measured and derived variables, as well as constants, are stored in a single table *Variable*. The values of those entities are stored in *Data* table. Whether *Data* is a relational table or a data structure residing in the operating memory of the server, depends on the implementation. Historical values of variables are stored in a separate *History* table.

Three tables store lists of variables. *ListS* stores derived variables that should be calculated after receiving a new value of each measured variable. *ListF* stores formal arguments of each expression. *ListA* stores actual arguments of each derived variable.

When the server is put into operation, its activities are triggered by arriving messages and by the flow of time. Each time a new transponder message arrives, the server calculates derived variables listed in table *ListS*. Panel messages are treated in the same way.

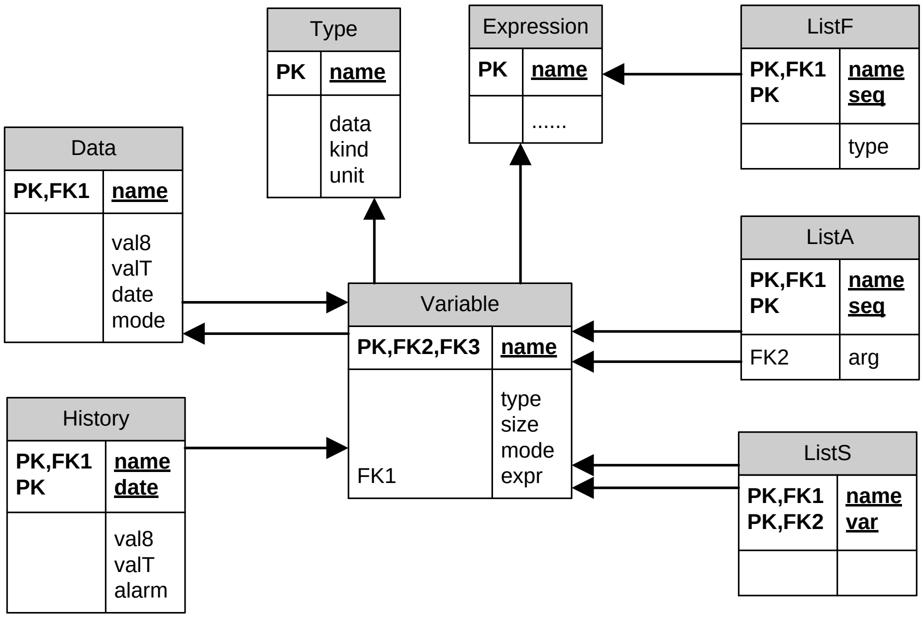

Fig. 3. Data model of SMARP server

Presentation of the plant status and operation is implemented by means of HTML web pages, which are generated automatically by the server using the specification of the plant structure and the current values of the process data stored in *Data* table.

A special mechanism for alarm handling exists in the server. The rules for raising an alarm and for handling the alarms are included into expressions that define derived variables. Each time a new value is calculated, then the alarm conditions are examined and if they are fulfilled, an alarm is raised. Alarm handling activities can display a message on a local touch panel or can send SMS or e-mail messages to the users. Displaying or sending a message is implemented by embedded server functions, which can be used within the expressions. When message is displayed on a touch panel, an operator has opportunity to enter a cause of alarm.

A lack of communication between transponder and server can be detected by the server and reported as an alarm. When communication is interrupted, there is a mechanism that allows server to order transponder to retransmit lost data.

In order to ensure that local clocks of all transponders and server are synchronized, SMARP provides a mechanism for time synchronization. It can be utilized, when there is no possibility to use standard mechanisms, such as NTP or Windows Time Service. This may happen when company policy doesn't allow for communicating external time server.

7 Conclusions and Future Work

The paper describes a new approach to production data monitoring and reporting. The monitoring and reporting system, called SMARP, is composed of a local plant transponder and a remote server located in the Internet. SMARP supports the manufacturing decision makers and provides aggregated process data that describes the effects of the plant operation, the overall equipment efficiency (OEE) and the causes of loses, such as accidents, damages and stoppages. The user can connect to the server through an arbitrary communication device, e.g. a laptop, a tablet or a mobile phone and inspect a production line status.

Currently, we are working on an application of SMARP system in an automated plant, which bottles and packs mineral water.

References

1. Boyer S.A.: SCADA Supervisory Control and Data Acquisition, ISA - International Society of Automation, USA (2010)
2. Bailey D., Wright E.: Practical SCADA for Industry, Newnes (2003).
3. OPC Foundation, http://www.opcfoundation.org/Downloads.aspx
4. Ozdemir E., Karacor M.: Mobile phone based SCADA for industrial automation. ISA Transactions 45 (1), 67-75 (2006)
5. Wallance D.: How to put SCADA on the Internet. Control Eng. 50 (9), 16–21 (2003)
6. Devices World, Malaysia, http://www.devicesworld.net/
7. Vipond Controls, Australia, http://www.vipondcontrols.ca/iscada/
8. Nicol G.T., XEXPR - A Scripting Language for XML, http://www.w3.org/TR/xexpr/
9. Brandes U., Eiglsperger M., Lerner J., GraphML Primer, http://graphml.graphdrawing.org/