Standardization of monitoring protocols for energy systems

Jean-Philippe Vanhulst, Jehan Snyers d'Atthenhoven
CHEROKEE International
Boulevard de l'Europe 135 – 1301 Wavre – Belgium
ETSI EE/EE2 Members
ETSI, 650 route des Lucioles- 06921 Sophia Antipolis – Valbonne, France

Abstract - **This paper is a more technical review of the new ETSI standard on control and monitoring of power and cooling systems for telecom sites and networks, delivered by the working group EE2. It focuses on the communication protocols used in this standard. This ES 202 336 standard is presented by Didier MARQUET, ETSI EE2 Chairman in the paper "New ETSI IP-XML power and cooling system monitoring and control interface standard"[1].**

Index Terms — **Supervision, Monitoring, Protocols, XML, SNMP, HTTP, Remote Management, Energy, DC System**

I. Introduction

Nowadays, telecom operators are using multiple standards to monitor their energy systems: dry alarms, proprietary protocols, SNMP, etc. The evolution of Distributed Network architecture calls for enhancements of control and monitoring. The use of dry alarms is difficult to manage and operators are asking for "all-IP" management solutions. Multiple vendors offer alternative solutions to achieve the goal. A lack of standardization in the industry has slowed down the adoption of new communication technologies. Standardization requires an agreement over the communication protocol and also over variables of measurement and control. This paper justifies the choice and provides more information about the chosen protocol during the standardization driven by ETSI EE2 committee (XML over TCP/IP).

II. Contributors

Please refer to the paper of D. Marquet [1] for the list of all the EE members which have contributed as observers and advisers to this work.

III. The ETSI organisation

Please refer to the ETSI organization presentation in referenced paper [1].

IV. The interoperability relays on the selected communication protocol

Interoperability is a key factor for the fast deployment of any network infrastructure and as such, it is obvious that proprietary protocols are not a good choice. The Simple Network Management Protocol (SNMP) is a widespread open protocol in the industry, commonly used for the supervision of IT networks. But SNMP has multiple drawbacks. For example, SNMP is often used over UDP which, unlike TCP, is an unreliable transport protocol. SNMP offers no guarantee of alarm delivery because a monitoring system is not able to know if an alarm/event message (SNMP Trap) was received or not by its management system. Routers, firewalls, and other IP-packet filtering devices often discard UDP packets. Moreover, depending on the number of routing nodes encountering on its way from a monitoring system to a management system, the maximum time to live (TTL) of a UDP packet can be exceeded, stopping the progression of the packet. Additionally with SNMP, when a trap is lost, it will not be transmitted again. (Using VPN and Wireless communication can also be critical). Another drawback comes from the fact that all the monitored objects have to be identified with the help of a MIB (Management Information Base), and this MIB system is not very flexible. Once a particular MIB is defined, it is difficult to make it evolve. Finally, MIBs of multiple vendors for similar equipment are almost never compatible together.

An innovative solution is to provide the data in a common syntax representation: the eXtensible Markup Language (XML). As defined by the World Wide Web consortium [2] XML is a simple yet very flexible text format. Originally designed to meet the challenges of large-scale electronic publishing, XML plays also an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. XML is designed to describe data and focus on what data is. Moreover, XML is a representation that is convenient for both Human-Machine and Machine-To-Machine communication.

This XML data representation needs a transport layer. For the time being, the better way to avoid compatibility and security problem but also firewalls and proxy filtering problems, is to communicate between applications over HTTP. This HTTP protocol communicates over the reliable well-known and widespread TCP/IP transport layer.

V. Quick XML overview

XML was designed to structure, store, and send information [3]. The following example shows a way to represent an equipment data, stored as XML.

```
<data>
   <name>Output Voltage</name>
   <value>54</value>
   <unit>Volt</unit>
   <type>measurement</type>
</data>
```

One can see that it is just pure information (the innertext) wrapped in XML tags. Some software must exist to send, receive, or display it in a user-friendly way. XML is a structured language: the <name> element is a child of the <data> element, as the <data> element is the parent of the <name> element.

The previous example has a correct syntax but represents a poor use of XML. XML elements can have attributes in name/value pair. These attributes must be quoted. A good practice rule is not to use attribute to store data, but rather for information about the data. According to this, the following structure is much better.

```
<data name=”Output Voltage” type=”measurement”
”unit=”Volt”>54</data>
```

The ETSI ES 202 336 standard defines standards XML tags and attributes allowing the easy parsing when retrieving information.

When a standard is written, it is possible to easily check the compliancy of any XML document toward this standard thanks to a XML schema. A XML schema describes the structure of a XML document. According to W3C standards [2], a XML schema:

- ✓ defines elements and attributes that can appear;
- ✓ defines which elements are child elements;
- ✓ defines the order of child elements;
- ✓ defines the number of child elements;
- ✓ defines data types for elements and attributes (string, double, integer, date, time, etc.);
- ✓ defines default and fixed values for elements and attributes.

A schema file will be included in ES 202 336 and soon be available from ETSI, allowing operators and vendors to quickly check the compliance of any equipment.

Some other W3C standards are related to XML. The eXtensible Stylesheet Language Transformation (XSLT) is used to transform a XML document into another XML document, or another type of document that is recognized by a browser, like HTML and XHTML. With XSLT, it is possible to add/remove elements and attributes to or from the output file. It is also possible to rearrange and sort elements, perform tests and make decisions about which elements to hide and display.

Another useful XML related language is XPath, used to find information in a XML document. It is convenient to navigate thought elements and attributes in a XML document. (For example, the xpath “/data/value” targets ‘54’ in the first XML example).

These two last technologies are not really involved in this ETSI standard. It has little sense to define a standard presentation of the information, or what must be filtered or not. Anyway, it is good to know how easy it is and will be for future developers to use the generated XML document (much easier than with SNMP).

VI. XML data structure described in ES 202 336 standard

As the structure of a XML document is really free, some rules are defined in the ETSI standard. This describes for example the order of the elements, the structure of an alarm and the place at which it shall be placed. Also, equipments and devices present on an energy site must be represented in a structured way. For example, while it is obvious that a rectifier is part of a DC system; the XML document must reflect this fact. Fig. 1 illustrates the hierarchy concept for a site. A dc system will always be embedded in an energy system, which will be owned by a site. All these elements and sub-elements have always a unique identifier attribute (id) allowing the distinction between same types of equipments.

A “status” attribute must be present for each equipment/system. This status can only have 3 values: “normal”, “alarms” or “unknown”. If the status is “alarms”, a “severity_type” attribute must be present, detailing the severity of the more severe active alarm.

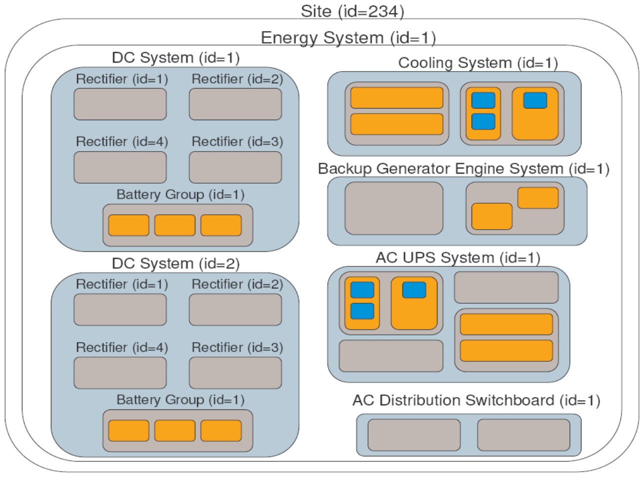


Fig. 1: Equipment hierarchy on a site

For each equipment, system, or subsystem, standard XML child nodes may exist once at most. Here follows an example of node called <an_equipment> with these standardized elements:

```
<an_equipment>
    <description_table>
        …
    </description_table>
    <alarm_table>
       …
    </alarm_table>
    <event_table>
        …
    </event_table>
    <data_table>
        …
    </data_table>
    <data_record_table>
        …
    </data_record_table>
    <configuration_table>
        …
    </configuration_table>
    <control_table>
        …
    </control_table>

    … sub equiments …

</an_equipment>
```

A. The description table (<description_table>)

This node contains multiple <description> elements, describing the equipment/system. The allowed attributes are an id, a name (in English), a group (allows filtering), a subgroup (second level of filtering), a unit, a datatype (allows parsing of the innertext), an info (for additional information on the description) and multiple name translations. Here follows 2 examples.

```
<description_table>
    <description id=”4” name =”Serial Number”
    group=”Manufacturer”>45623-5F-EG</description>
    …
    <description id=”7” name =”Max Output Power”
    group=”Manufacturer”  subgroup=”Specifications”
    unit=”watt”>850</description>
</description_table>
```

B. The alarm table (<alarm_table>)

This node contains multiple <alarm> elements, each describing the alarm itself and the status of this alarm. The principal allowed attributes are an id, an active status (true or false in order to know if the alarm is active or not), a name, a severity type (critical, major, minor, warning or information), a severity level (value from 0 to 0), a start time and a stop time. Here follows two examples.

```
<alarm_table>
    <alarm id=”1” active=”false” name=”DC bus Low”
    severity_type=”major” severity_level=”5”/>
    …
    <alarm id=”3” active=”true” name =”Mains Fail”
    severity_type=”major” severity_level=”5” start_time=”2006-12-
    17T18:23:12Z”/>
</alarm_table>
```

C. The event table (<event_table>)

This node contains multiple <event> elements, each describing an event which appended at the corresponding equipment/system. The main allowed attributes are an id, a type (alarm start, alarm stop, information), a date and time, a severity type, a severity level, an alarm id (according to alarm table), additional information. The event sentence is the innertext of the XML element. Here follows 3 simple examples.

```
<event_table>
    <event id=”1” type=”information”  datetime=”2006-12-
    17T18:23:12Z”>Equipment started</event>

    <event id=”2” type=”alarm_start” severity_type=”major”
    severity_level=”5” alarm_id=”2” datetime=”2006-12-
    17T19:25:12Z”>Alarm appeared: Mains fail</event>

    <event id=”3” type=”alarm_stop” severity_type=”major”
    severity_level=”5” alarm_id=”2” datetime=”2006-12-
    17T20:25:12Z”>Alarm disappeared: Mains fail</event>
</event_table>
```

D. The data table (<data_table>)

This node contains multiple <data> elements, which provides measurements, states and calculated values related to the equipment/system. The main allowed attributes are an id, a name, a group, a subgroup, a type (measurement, calculated_value, etc.), a unit, an accuracy, a measurement type (peak, rms, max, min, etc.), a datatype (decimal, boolean, integer, etc.), a date and a time (when was the data calculated or measured ?), an additional information, some translations of the data name, etc. The innertext is the data value. Here follows an example.

```
<data_table>
    <data id=”1” name=”Output Voltage” type=”measurement”
unit=”volt” accuracy=”1%” format=”xs:decimal” datetime=”2006-12-
    17T18:23:12Z”     name_FR=”Tension de sortie”>54</data>
</data_table>
```

E. *The data record table (<data_record_table>)*

This node contains multiple <data_record> elements, which provides historic or statistics of some data present in the data table. Please refer to the standard as it is not possible to show such information in this kind of paper.

F. *The configuration table (<config_table>)*

This node contains multiple <config> elements, which provides a way to know and configure parameters of the equipment/system. The main allowed attributes are an id, a group, a subgroup, a type, a unit, a datetime (date and time of the last configuration change), additional information, and some name translations.

G. *The control table (<control_table>)*

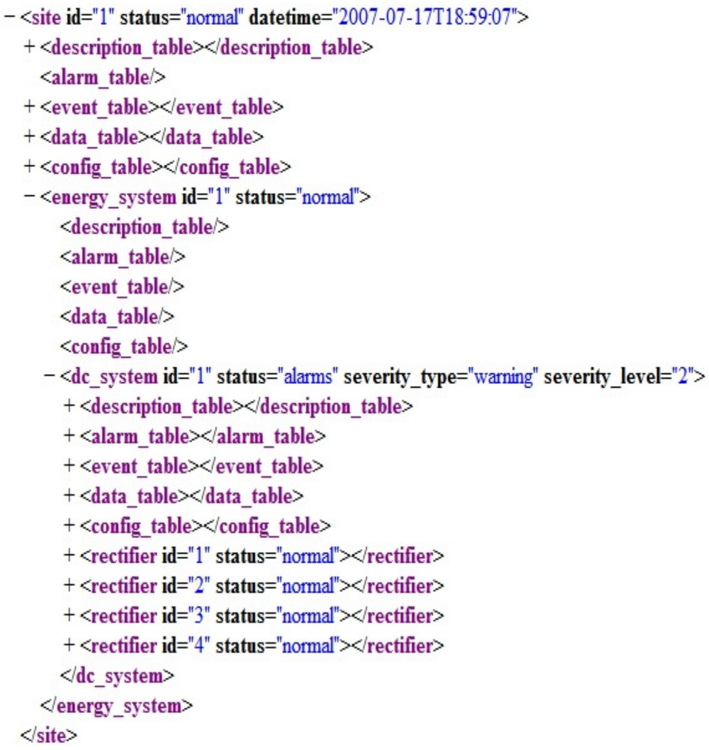
This node contains multiple <control> elements which provides a way to know the permitted control operations on the devices and to execute them.

H. *Example of XML document*

Fig. 2 is a typical example of a part of XML document containing many descriptions, alarms, events, data and configurations related to a given site. The root element of a document generated by a site monitoring must be “site”. The “datetime” attribute gives information about the time at which the document was generated.

VII. Exchanging XML data

To exchange XML data between network elements, the communication must be exclusively based on open protocols to avoid any problems of interoperability. Many XML compatible technologies exist to retrieve information, to configure, to send alarms and events. The final user has to decide the one which can be easily integrated in his management system. The communication protocol is not defined in this ETSI standard. However, several protocols are introduced and/or illustrated.



```
- <site id="1" status="normal" datetime="2007-07-17T18:59:07">
  + <description_table></description_table>
    <alarm_table/>
  + <event_table></event_table>
  + <data_table></data_table>
  + <config_table></config_table>
  - <energy_system id="1" status="normal">
      <description_table/>
      <alarm_table/>
      <event_table/>
      <data_table/>
      <config_table/>
    - <dc_system id="1" status="alarms" severity_type="warning" severity_level="2">
      + <description_table></description_table>
      + <alarm_table></alarm_table>
      + <event_table></event_table>
      + <data_table></data_table>
      + <config_table></config_table>
      + <rectifier id="1" status="normal"></rectifier>
      + <rectifier id="2" status="normal"></rectifier>
      + <rectifier id="3" status="normal"></rectifier>
      + <rectifier id="4" status="normal"></rectifier>
      </dc_system>
    </energy_system>
  </site>
```

Fig. 2: Example of XML document related to a site

A. *Retrieving XML document with HTTP get*

Each monitoring system can act as a Web server, holding one or more XML documents. Each existing document is within the HTTP standard referred as a resource. Each resource is identified by a unique resource identifier known as a URI (Uniform Resource Identifier). An example of URI is “http://10.23.45.98/site.xml”. The first part of the URI is always the IP address of the site. If a hostname is defined, the IP address can be replaced as follow: http://my_site_hostname/site.xml.

A request to such an URI will result in a response message from the server with information about the status of the request and, in applicable cases, the XML document requested.

An example of HTTP get request sent by the Firefox browser retrieving the “site.xml” file looks like:

```
  GET /site.xml?description_table=false HTTP/1.1
  Host: 10.23.45.1
  User-Agent: Mozilla/5.0 (Windows; U; Windows NT
5.1; fr; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4
  Accept:
text/xml,application/xml,application/xhtml+xml,text
/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
  Accept-Language: fr,fr-fr;q=0.8,en-
us;q=0.5,en;q=0.3
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
  Keep-Alive: 300
  Connection: keep-alive
```

The monitoring system will then respond with:

```
HTTP/1.1 200 OK
Server: ServerName
Content-Type: text/xml
Content-Length: 28564
```

→ Followed by the generated XML document bytes.

Some parameters can be passed to the request URI in order to retrieve only parts of the full site XML document.

Parameter name	Value	Description
description_table	true/false	Define if the description table must be included in the generated xml document (at each level of hierarchy)
alarm_table	true/false	Define if the alarm table must be included in the generated xml document (at each level of hierarchy)
event_table	true/false	Define if the event table must be included in the generated xml document (at each level of hierarchy)
data_table	true/false	Define if the data table must be included in the generated xml document (at each level of hierarchy)
data_record_table	true/false	Define if the data record table must be included in the generated xml document (at each level of hierarchy)
config_table	true/false	Define if the config table must be included in the generated xml document (at each level of hierarchy)
level	0, 1, 2, etc	Define the maximum level of hierarchy. 0 will retrieve only the site level, 1 will retrieve site and energy system level, etc.
option	0, 1, 3, etc	Generate the XML document according to a specific option (defined by the manufacturer)

The order of the parameters is free, and none is mandatory. The default XML document, when no parameter is provided, is defined by the manufacturer.

For example, if we want to retrieve the data table and the alarm table up to the third level of hierarchy, the URI will be:

http://the_site_ip/site.xml?description_table=false&alarm_table=true&event_table=false&data_table=true&data_record_table=false&config_table=false&level=3

Once the XML file is downloaded, with commonly used W3C standards like XSLT, CSS, HTML, XHTML, AJAX, JavaScript, etc., a user-friendly web page can be generated, including the gathered data.

B. Web services (SOAP protocol for example)

"Web Services" represents probably the most flexible way to exchange data. This technology enables computer systems on any platform to communicate over corporate intranets, extranets, and across the Internet with support for end-to-end security and reliable messaging. These are based on a core of standards that describe the syntax and semantics of software communication.

The more widespread Web Service protocol is "SOAP". It is a simple XML based protocol that let applications exchange information over HTTP. This protocol is standardized by the World Wide Web consortium [2]. This standard was initiated by companies like Commerce One, Compaq, Developmentor, HP, IBM, IONA, Lotus, Microsoft, and SAP [3].

A SOAP message is an ordinary XML document containing at least an "Envelope" that identifies the XML document as a SOAP message and a "Body" element that contain call and response information.

Here follows an example of SOAP request/response in order to change the site name description on a site. The function called on the monitoring over SOAP Web Services is "SetValue". The parameters are a path to the target value, and the new value.

Request:

```
  POST /webservices.asmx HTTP/1.1
  Content-Type: text/xml; charset=utf-8
  SOAPAction: "http://tempuri.org/SetValue"
  Host: 10.23.45.1
  Content-Length: 348
  Expect: 100-continue

  <?xml version="1.0" encoding="utf-16"?>
  <soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelop
e/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Body>
      <SetValue xmlns="http://tempuri.org/">
<path>/site/1/description_table/description/1</path
>
  <value>A Site Name</value>
      </SetValue>
    </soap:Body>
  </soap:Envelope>
```

Response:

```
  ResponseCode: 200 (OK)
  Content-Length:378
  Content-Type:text/xml
  Server:ServerName

  <?xml version="1.0" encoding="utf-16"?>
  <soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelop
e/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
    <soap:Body>
      <SetValueResponse
xmlns="http://tempuri.org/">
        <SetValueResult>ok</SetValueResult>
      </SetValueResponse>
    </soap:Body>
  </soap:Envelope>
```

Any function can be created that way, with specific parameters and specific responses. Proceeding that way with SNMP is a real nightmare.

Web Services can be implemented both on the monitoring systems and on the management system. Then, the monitoring system is able to post events by calling a SOAP function on the management system.

C. *Some other protocols*

As the transport layer for XML is really free, it is possible to use many other transport layers:

- ✓ XML documents can be retrieved or sent trough FTP client/server architecture;
- ✓ XML documents can be posted on a Web server periodically, or only in case of problem;
- ✓ XML documents can be mailed for critical problems;
- ✓ XML documents can be exchanged trough a Telnet socket;
- ✓ If security is required, the usual secured protocols can be used: HTTPS, FTPS, SSH, etc.;
- ✓ … Possibilities are almost unlimited.

VIII. Bandwidth usage

A common drawback of these recommended protocols is that the requested bandwidth is higher than SNMP. This is true in the absolute. Indeed, with SNMP, as UDP is not absolutely reliable, a permanent polling is necessary for a save network supervision. With the proposed monitoring approach, polling is not necessary as the communication is reliable. Also, in order to retrieve all the data related to a system (for deep diagnostic), the SNMP protocol needs many request/response operation. With XML, with only one request, one can retrieve a large file containing everything (system description, list of alarms, log of events, data, data records, etc).

Moreover, with the constant increase of bandwidth avaibility, the bandwidth consumption problem becomes less and less critical. Depending on the XML document retrieved, from some kilo octets to hundreds of kilo octets will be retrieved, which is still comparable to the size of a small image.

IX. Conclusion

The use of XML over TCP/IP for site and system monitoring has many advantages over dry alarms, proprietary protocols, or the classical SNMP. SNMP was a good solution years ago, but this protocol is not flexible enough for new telecom all-IP managing networks.

During next months, vendors will certainly begin to integrate these new supervision techniques, according to ETSI recommendations. It will ease the monitoring of energy systems from multiple vendors and will provide a great tool for the remote execution of more complex functions.

As of today, as explained by D. Marquet [1], only the core XML tags are defined, as overviewed in this paper. In a near future, standard tags will be defined for all the equipments present on telecom site (DC System, Air Conditioning, AC distribution, etc.). Recommendations are also given over the necessary information, data, configuration, etc. for these systems.

References

[1] Marquet D. New ETSI IP-XML power and cooling system monitoring and control interface standard, Intelec 2007 Rome
[2] World Wide Web Consortium Website: http://www.w3c.org
[3] W3 Schools, the best things in life are free: http://www.w3schools.com
[4] Final Draft ETSI ES 202 336-1, ETSI Standard.