ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2011. No. 1(107)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 120 SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Extension of the Common Information Model with Virtual Meter

S. Vukmirović, A. Erdeljan, I. Lendak, D. Čapko
Computer and Control Department, University of Novi Sad,
Trg Dositeja Obradovica 6, Novi Sad, Serbia, phone: +381 64 13 23 706, e-mail: srdjanvu@uns.ac.rs

Introduction

The demand for electricity has been steadily growing over the last few decades. Since traditional energy resources are limited, energy efficiency has become a fundamental concern at all levels of the power system: at the generation and transmission (Energy Management Systems – EMS), power distribution (Distribution Management Systems – DMS) and eventually at the level of the energy consumer. The development of advanced smart meters allows individual consumers to take part in the process of energy saving. While the first generation of such meters was only able to measure consumption, the latest models allow the measurement of various consumption parameters. Moreover, these meters allow some consumer devices to be turned off if necessary. Thusly, meters cease to be passive devices only: they become an important part of a complex and general supervisory and control system [1]. It is expected that these systems will allow energy savings during critical consumption peaks.

The primary goal of the Smart Metering system is to gather data from remote power consumption meters and transfer this information to the general supervisory and control systems of power utilities. The network of meters is a large distributed system incorporating hundreds of thousands (or even millions) of devices. When developing software for such large scale systems, it is extremely important to create highly reliable and robust software architectures using new technologies. As the Smart Metering software handles sensitive data which can be misused (i.e. energy theft, consumer identity theft, unauthorized load shedding), it is necessary to incorporate advanced security techniques in all phases of design. They are expected to work without interruption (twenty four hours, seven days a week – 24/7) and therefore all components must be exceptionally reliable and fault tolerance should be taken into consideration at all levels. Another important aspect to be considered in these systems is the fact that the Smart Metering systems have to exchange data with other subsystems and applications used by power utilities, which makes the clear and concise definition of integration interfaces very important.

The CIM (Common Information Model) [2] is an abstract model representing all important objects of an electrical power system in the form of UML (Unified Modeling Language) diagrams [3, 4]. Ensuring a standard way for representing resources of an electrical power system as classes, objects, attributes and associations between them, the CIM enables a seamless integration of applications independently developed by different software vendors, the integration of whole disparate energy management systems, and/or the integration of an EMS with other systems. The CIM was initially built for the purpose of EMS, as well as for controlling power production and supervising transmission networks. However, for the purposes of the application in distribution networks, the CIM could benefit from certain modifications and enhancements. The main characteristic of distribution networks is their huge dimensionality, which is multiple times bigger than in the case of transmission networks for which the CIM was originally crafted. Besides, the CIM was developed primarily as a data exchange model, not as an internal data representation model.

The large number of meters that are anticipated to be managed by the Meter Data Management (MDM) tools impose the large dimensionality of the problem. This kind of dimensionality creates problems in memorizing meter readings data and in attaining the acceptable speed in processing meter reading results inside distributed MDM applications. Software integration, from the perspective of power applications, aims to facilitate these kinds of applications by exchanging model data with each other. Generally, this may be achieved by either assembling the pertinent information into standard CIM/XML documents representing the unit of exchange in the Enterprise Service Bus (ESB) architecture, or the relevant information can be stored in a CIM based database for common access. In both cases the granularity of the data exchange itself is usually appropriate only at an application level, but not at a finer grained component level.

This paper consists of six parts. The second part includes an overview of related works, and the third part describes the architecture of an MDM system. Part IV describes the proposed way of modeling the Virtual Meter in the CIM. Part V presents the test results, whereas part VI concludes the paper.

Related works

Although the concept of smart metering is relatively novel, considerable research efforts have been invested in this area. One of the key components regarding the implementation of this concept in power distribution systems is Advanced Metering Infrastructure (AMI) [5]. Some authors estimate that one can save 3% on system maintenance merely by introducing AMI systems. These savings come from the reduction in maintenance costs and faster system restoration after outages. After the integration of AMI systems with other subsystems, we can expect up to twice as much saving [1]. Key aspects of Smart Metering systems are collaboration and integration, which will greatly impact the adoption of the products in the market [6]. Because of the above mentioned facts, in this phase of the development of AMI systems their fast, simple and secure integration with other systems is extremely important. It is necessary to choose well tested and proven industry standards as a basis for this integration. There are works which show various Smart Metering system architectures, but they are either highly specialized and solve only a single problem or only show a high level architecture barely touching upon integrations and hardly defining data models and integration interfaces [7, 8].

Apart from monetary and operational gains from Smart Metering, the proposed architecture also brings social and environmental gains. Although the calculation of social gains can not be accurate, they are estimated to make a significant part of total saving [9, 10].

Architecture proposed in this paper can have especially great impact in systems with Renewable Energy Resources where a large number of independent Energy sources have to be processed quickly [11].

The CIM was originally developed for Energy Management Systems (EMS). Although it has been used for years now, it has not yet been widely accepted in the field of DMS software mainly because there was no clear integration methodology defined [12]. Fortunately, supporting documents issued at the end of 2009 describe the architecture of the system and possible integration methodologies, which will hopefully lead to a faster acceptance of the CIM [13]. The CIM describes equipment and their connections as used in EMS/DMS. As the model is very detailed and contains a large number of various types, it is divided into several packages containing both electrical and non-electrical concepts. All objects inherit a class (*IdentifiedObject*) from a common base and all ensuing class inheritance is logical and based on the appearance and use of objects encountered in real power systems. In this manner it was possible to maintain a well-organized and easy-to-understand model, although it contains hundreds of classes. There are papers discussing XML based data exchange which would allow an optimal speed and ease of extension [14, 15]. Others discuss various ways to adapt the CIM for the use with Web Services and Enterprise Service Buses (ESB) [16, 17], but these solutions are still in their early phase of development.

Meter Data Management Architecture

One of the most important goals of Smart Metering systems is to gather measurement values from distant measurement devices (meters) and integrate them into a complex and unified system for acquisition and control in power distribution systems. With the introduction of new meters it is now possible to control consumption per individual consumer. This evolution makes distribution management systems more complex. The integration with other subsystems and a clearly defined data flow between them becomes increasingly more important.

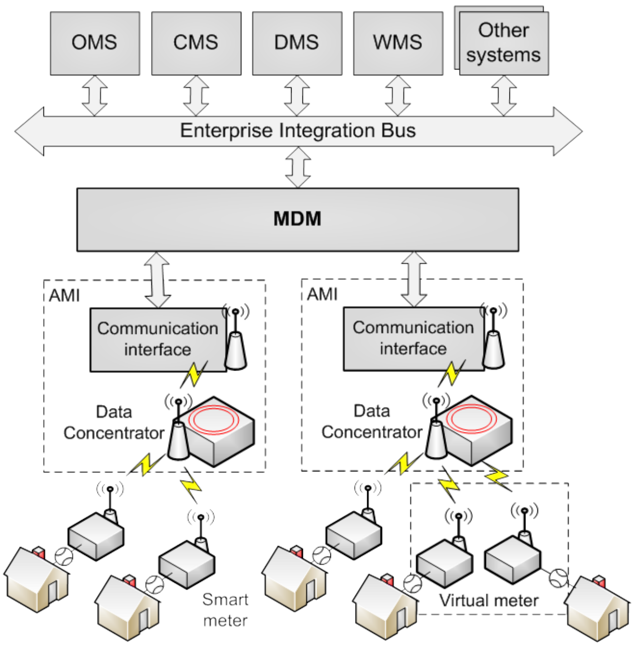

Fig. 1. Open Smart Metering architecture

The vendor of measurement devices usually provides an infrastructure for centralized measurement acquisition through concentrators gathering signals from multiple meters. These systems, which are able to gather measurements from meters built by a single vendor, are called Advanced Metering Infrastructure systems. As a single utility controlling a power distribution system can purchase different meters from different vendors, it has become necessary to develop central systems which would gather data from various AMI systems operated by a single utility. These central systems are called Meter Data Management (MDM) systems, which, besides gathering data, also perform data processing.

One of the most important tasks of the MDM is processing large volumes of data and preparing it for use by other subsystems of Smart Grid. Virtual Meter is designed with this goal in mind; it will reduce time of processing of metering data by preprocessing readings by predefined criteria. This way some statistical functions could be applied to logically connected meters and prepare group statistics for the analysis.

As the values measured by meters become more and more important for other subsystems (Outage Management Systems – OMS, Customer Management System - CMS, Distribution Management System - DMS and Work

Management System - WMS) their integration with MDMs can be implemented through an Enterprise Service Bus (ESB) using the data model and interfaces proposed in this paper. The architecture of the proposed system is shown in Fig. 1.

Modeling Virtual Meter in CIM

The CIM defines object models for power generation, transmission and distribution systems. Although the CIM has hundreds of classes for modeling various components of a power system, in this paper we will focus on a handful of classes which are interesting for Smart Metering systems. The base class in the CIM is the *IdentifiedObject*: it contains a unique identifier, name, description and path. *Asset* class represents a tangible resource of the utility, including power system equipment, cabinets, buildings, etc. [18]. The CIM models a meter object by the *MeterAsset* class (which is derived from the Asset class), whose functions are metering, load management, connect/disconnect, accounting functions, etc. The *MeterReading* groups a set of values obtained from a meter. A single measured value is modeled by the Reading class, which is derived from *MeasurementValue*, and it contains a value, the precise time of measurement and its quality.

A Virtual Meter is an abstraction that may consists of multiple meters. Through these it is possible to create virtual meters which measure consumption in a part of the distribution network (summing the consumption measured by meters in that part of the network). Data processing functions can handle virtual meters just as they handle regular meters. It is possible to validate and estimate data or send data into other subsystems.

The Virtual Meter can organize meters by a variety of criteria:

- By the customer, if a single customer has multiple meter devices;

- By the type of meters (or meter vendor) if a power distribution utility needs a report by this criterion for some business logic;

- By the type of consumers (business, industrial, residential, etc.) for the purpose of validation and estimation or similar business logic;

- By the geographical area, thus creating reports for a specific area;

- By the user responsible for a group of meters, for the purpose of employer activity audit;

- By the distribution equipment, e.g. by Low Voltage Transformer where meters are connected. In this way utility can monitor power efficiency and energy theft.

By introducing the Virtual Meter a certain amount of preprocessing could be done on a scheduled basis in order to prepare data for processing. In such a way some statistical function could be applied to measured values and most often it will be a sum of measured values, but it can also be a mean, a maximum or a minimum. Software architecture presented as part of the solution in this paper has a subsystem that can execute this kind of business logic as a service component.

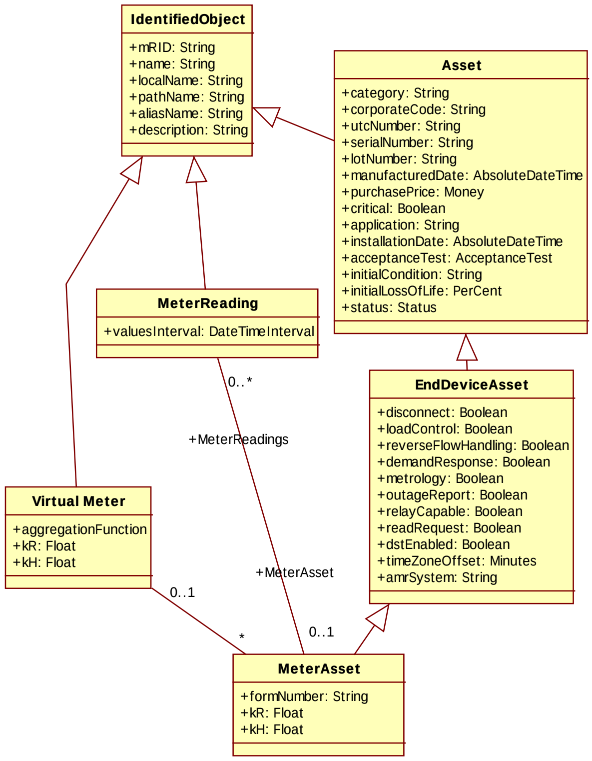

Fig. 2. CIM model with Virtual Meter

Fig. 2 shows the CIM model with a new class *VirtualMeter* and its connections with other relevant classes of the CIM model. The *VirtualMeter* is derived from the *IdentifiedObject* class (the base class of CIM model).

IdentifiedObject class is abstract and only contains attributes used to reference the object either by a user or in software. The attributes of *IdentifiedObject* include *mRID*, which is the master resource identifier that should be a globally unique identifier of objects; the mRID does not have to be human-readable. This identifier is generally intended to be used by software systems.

The attributes *name*, *description*, *aliasName*, and *pathName* are intended for providing identifiers that are human-readable. It is common for names of objects within a utility not to be unique due to historical naming conventions, the results of mergers and acquisitions, and the inability of other software systems to manage uniqueness. For these reasons, there are no requirements that these names be unique.

Attributes that are specific for the *VirtualMeter* class are:

- *aggregationFunction* (of the type *AggregationFunction*), which can be used to specify which aggregation function is used for the virtual meter calculation. The available options are: *none, sum, min, max* and *mean.*

- *kR* - Display multiplier used to produce a displayed value from a register value.

- *kH* - Meter kh (watt-hour) constant. It is the number of watt-hours that must be applied to the meter to cause one disk revolution for an electromechanical meter or the number of watt-hours represented by one increment pulse for an electronic meter.

The Virtual Meter has one association that connects it with the MeterAsset class:

- *MeterAssets* - All meter assets that belong to this virtual meter.

Since CIM extension is XML base, Fig. 3 shows the schema (XSD) for VirtualMeter class.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
targetNamespace="http://iec.ch/TC57/2009/VirtualMete
r#" xmlns:m="http://iec.ch/TC57/2009/VirtualMeter#">

<xs:element name="VirtualMeter"
type="m:VirtualMeter"/>
<xs:complexType name="VirtualMeter">
  <xs:sequence>
            <xs:element name="EndDeviceAsset"
             type="m:EndDeviceAsset"
             minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="aggregationFunction">
  <xs:restriction base="xs:string">
    <xs:enumeration value="min"/>
    <xs:enumeration value="max"/>
    <xs:enumeration value="mean"/>
    <xs:enumeration value="sum"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="kR">
  <xs:restriction base="xs:float"/>
</xs:simpleType>

<xs:simpleType name="kH">
  <xs:restriction base="xs:float"/>
</xs:simpleType>
</xs:schema>
```

Fig. 3. XML schema for Virtual Meter

An example of a Virtual Meter XML message is shown in Fig. 4. Apart from attributes of Virtual Meters there are two lists of elements. The fist list contains all meters that belong to this Virtual Meter. This list could be omitted for performance reasons if only meter readings are needed for the Virtual Meter. The second list in the Virtual Meter contains a list of readings (or an aggregation of reading). This list will not be used for methods that are used with the Virtual Meter.

```
<?xml version="1.0" encoding="UTF-8"?>
<m:VirtualMeter
xsi:schemaLocation="http://iec.ch/TC57/2010/VirtualM
eter# VirtualMeter.xsd"
xmlns:m="http://iec.ch/TC57/20107/VirtualMeter#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">

<m:mRID>47474</m:mRID>
<m:name>Virutal Meter 1</m:name>
<m:agregationFunction>sum</m:agregationFunction>

<m:MeterAsset>
     <m:mRID>6423144</m:mRID>
     <m:name>Meter202332</m:name>
</m:MeterAsset>
<m:MeterAsset>
     <m:mRID>6468822</m:mRID>
     <m:name>Meter203168</m:name>
</m:MeterAsset>
<m:Reading>
     <m:timeStamp>2009-12 17T09:30:47</m:timeStamp>
     <m:value>3.12359E0</m:value>
</m:Reading>
</m:VirtualMeter>
```

Fig. 4. XML payload for Virtual Meter

A small drawback of introducing a new class in the CIM model is that this new class makes the implementation of the CIM compliant server somewhat complicated but performance tests in this paper will show that this effort is worthwhile. The creator of the CIM, the Electric Power Research Institute (EPRI), defines function formats for exchanging the CIM objects [19]. This is also something that is implemented for the *VirtualMeter* class. Since this object has a very straightforward meaning, the implementation of these functions is not complicated. Finally, there is the cost of invoking these new functions in runtime (adding a new virtual meter, updating the existing one and deleting unnecessary virtual meters). Once again we think that this function call will be used rarely since virtual meter grouping data will not be changed often. On the other hand, a significant performance bust justifies this meter maintenance cost.

Test results

The performance testing of the proposed extension model was carried out on real meter readings from a pilot project. Meters are configure to read just one value – interval power consumption. Meter power consumption values are read in 15 minute intervals during a period of one month, so there are 96 reads per meter per day, which totals in 2880 reads per month. The period of one month is chosen because most reports and validation rules are done based on a month time interval. We assume that some values are not read due to communication problems, but most of the values are stored in the database.

The pilot project had approximately 100 000 meters, which gives us the opportunity to test the Virtual Meter approach since it is very unlikely that a single Virtual Meter will have more that 100 000 meters in it. Two test scenarios are designed to show two most significant aspects of system integration. The first test, named Execution speed performance test, shows the time needed to execute the aggregation function on meters in *VirtualMeter*. In the second test, Exchange message size, the message payload size is analyzed.

A. Execution speed performance test. The first test is designed in order to measure the speed of the execution of the reporting process for Virtual Meter. One of the most important reports that customers ask from the MDM system is Power Audit. This report shows the difference between the sum of metering data from the meters connected to a single low voltage transformer and the metering data from the transformer.

For testing purposes Virtual Meters are created with an appropriate number of Meters, ranging from 10 to 100000. The aggregation function is the sum, since we anticipate that this function will be the most commonly used function. Table 1 shows testing results: the first column shows the number of meters in the virtual meter and the second column shows the time needed for the execution of Reading of data without the concept of Virtual Meter. The third column shows execution time with the Virtual Meter concept, but when the processing of aggregation function is done I n function call. This approach has improved performances because less data is transferred through web services. The last column shows test results in the environment with preprocessing of data. In this case the improvement of preferences is the most significant.

Table 1. Speed of execution performance test

Number of meters	Execution time without Virtual Meter [s]	Execution with Virtual Meter without preprocessing [s]	Execution time with Virtual Meter with preprocessing [s]
10	0.9	1	0.1
100	6	2	0.1
500	30	9	0.1
1000	60	12	0.1
5000	300	25	0.1
10000	600	48	0.1
50000	3000	55	0.1
100000	6000	58	0.1

Fig. 5 shows testing results described in Table 1. It is easy to see that even without preprocessing getting measurement data from an MDM system is significantly improved with the introduction of Virtual Meter. Improvement is greater for virtual meters with a larger number of meters. The most significant speed performance impact is achieved if the preprocessing of data is done; we anticipate that this will be a fairly common case since most of the systems that use MDM data will use periodical data for a day, week or month so preprocessing could be done in a predefined time period.

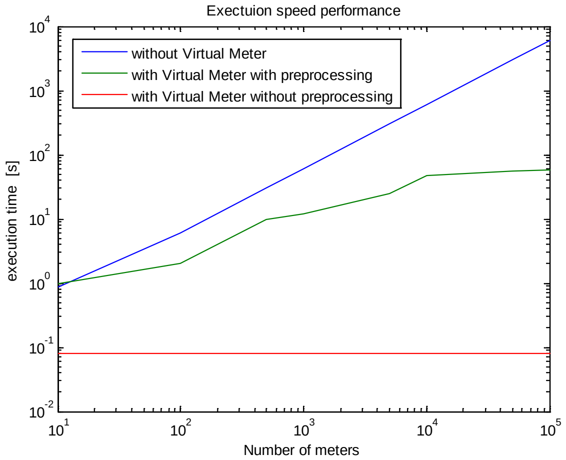

Fig. 5. Speed or execution performance test

B. Size of export file test. The second test of performance for Virtual Meter was the size of export file that is exchanged between subsystems. The XML format for Virtual Meter has been described previously. This test shows an enormous change of the size of exported data. Once again there are three columns with data: the first column lists the size of the exported file without virtual meter, the second column lists values with virtual meter plus definition of meters, and the third column shows results when just Virtual Meters are sent. We have anticipated that Virtual Meter configuration will not change very often and that in most cases only the consumption data per Virtual Meter will be exchanged. Fig. 6 shows testing results described in Table 2.

Table 2. Size of exported file

Number of meters	Message size without Virtual Meter [MB]	Message size with Virtual Meter and Meter List [MB]	Message size with Virtual Meter, without Meter List [MB]
10	3.8	0.5	0.5
100	40	0.5	0.5
500	200	0.5	0.5
1000	400	0.5	0.5
5000	2000	0.6	0.5
10000	4000	0.8	0.5
50000	20000	2.5	0.5
100000	40000	4.5	0.5

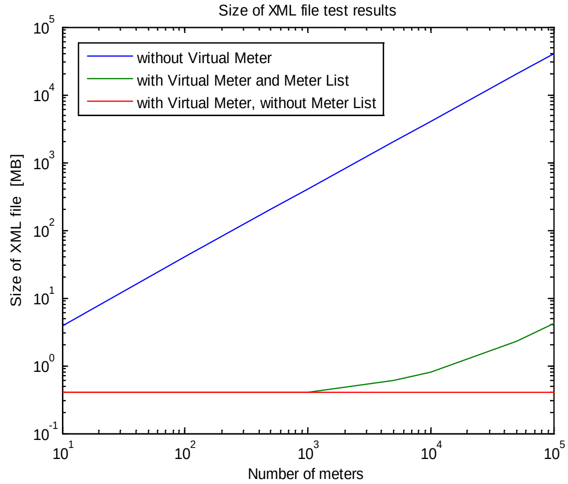

Fig. 6. Size of exported file

Conclusions

This paper proposes the extension of the CIM standard with Virtual Meter. Virtual Meter could be used for preprocessing of logically connected meters, which could be logically connected by customer data, an operational connection, a meter type or a manufacturer. With Virtual Meter preprocessing of data could be done centrally in the MDM system and after that all other Smart Grid components could use this data. In this manner both the processor power and the communication channel usage performances could be improved.

A novel architecture for Smart Metering systems, which allows rapid development, scalability and easier maintenance, is suggested in this paper. With this architecture the concept of Virtual Meter could be fully exploited.

Two performance tests were designed to show that both the speed of execution of reading and the size of export file could be improved with the proposed model extension.

The drawback of this extension is that by introducing a new class, the MDM system becomes slightly more complicated. Also a small number of functions for maintaining data about Virtual Meters should be implemented. But as performance tests show, this additional work for maintenance is highly profitable.

The proposed extension and novel architecture were implemented within a commercial Meter Data Management system, thereby proving its worth.

References

1. **Tram H.** Technical and Operation Considerations in Using Smart Metering for Outage Management // Transmission and Distribution Conference and Exposition. – T&D. IEEE/PES, 2008. - P. 1-3.
2. **Electric Power Research Institute (EPRI),** Common Information Model (CIM): CIM 10 Version. – Palo Alto, CA, 2001. – 1001976.
3. **Booch G., Jacobson I., Rumbaugh J.** The Unified Modeling Language User Guide. – Addison-Wesley, 1999.
4. **Turcu C., Prodan R., Cerlincă M., Turcu C., Cerlincă T.** RFID@B2B a Powerful Enabler of Business Transformation // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 5(93). – P. 59–64.
5. **Tram H.** Using AMI to realize the Smart Grid // Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, IEEE. – Pittsburgh, 2008.
6. **Vukmirovic S., Erdeljan A., Capko D., Lendak I.** Smart Metering architecture for power distribution efficiency // Second regional conference industrial energy and environmental protection in southeastern European countries. – JZlatibor, Serbia, 2010.
7. **Fischer R., Schulz N. N., Anderson G. H.** Information Management for an Automated Meter Reading System // 2000 American Power Conference. – Chicago, 2000.
8. **Pamakštis V., Ragauskas A., Kaluževičius, J. Petkus V., Bagdonas A., Borodičas P.** Advanced Energy Metering and Control System // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008 – No. 7(87). – P. 93–98.
9. **Neenan B., Hemphill R. C.** Societal Benefits of Smart Metering Investments // The Electricity Journal, 2008. – Vol. 21. – Iss. 8. – P. 32-45.
10. **Loveday D. L., Bhamra T., Tang T., Haines V.J.A., Holmes M. J., Green R. J.** The energy and monetary implications of the ‘24/7’ ‘always on’ society // Energy Policy, 2008. – Vol. 36. – Iss. 12.– P. 4639-4645.
11. **Kunicina N., Galkina A., Zhiravecka A., Chaiko Y., Ribickis L.** Increasing Efficiency of Power Supply System for Small Manufactures in Rural Regions using Renewable Energy Resources // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 8(96). – P. 19–22.
12. **Lendak I., Erdeljan A. & Perlić Z.** Architectural issues in CIM to relational model conversion // 16th DAAAM International Symposium "Intelligent Manufacturing & Automation: Focus on young researchers and scientists". – Opatija, Hrvatska, 2005.
13. **Electric Power Research Institute** An Introduction to the CIM for Integrating Distribution Applications and Systems. – 2008.
14. **Popović D., Varga E., Perlić Z.** Extension of the Common Information Model with a Catalog of Topologies // IEEE Transactions on Power Systems, 2007.
15. **McMorran A. W., Ault G. W., Elders I. M., Foote C. E. T., Burt G. M., McDonald J. R.** Translating CIM XML Power System Data to a Proprietary Format for System Simulation // IEEE Transactions on Power Systems, 2004. – Vol. 19. – No. 1.
16. **Vukmirovic S., Lukovic S., Erdeljan A., Kulic F.** Software architecture for Smart Metering systems with Virtual Power Plant // The 15th IEEE Mediterranean Electrotechnical Conference. – Valletta, Malta, 2010.
17. **Electrotechnical Commission (IEC) Task Group 13** IEC 61968-11 Ed.1: Application integration at electric utilities - System Interfaces for Distribution Management (Part 11: Common Information Model (CIM) Extensions for Distribution).
18. **Electric Power Research Institute** Enterprise Service Bus Implementation Profile. – 2009.

Received 2010 07 06

S. Vukmirović, A. Erdeljan, I. Lendak, D. Čapko. Extension of the Common Information Model with Virtual Meter // Electronics and Electrical Engineering. – Kaunas: Technologija, 2011. – No. 1(107). – P. 59–64.

This paper presents an extension of the CIM (Common Information Model) with a definition for Virtual Meter. Virtual Meter is a logical group of Meters that can be used for preprocessing the data for the performance purposes. This extension has been implemented by defining an additional set of classes for Virtual Meters. The extended CIM was tested on real meter reading data from power distribution utility. Performance metrics of the model, taken into account at the time of the evaluation were: the size of the CIM/XML (eXtensible Markup Language) document containing exported data about the Virtual Meter, and the speed of processing the Meter Readings data in advanced Meter Data Management (MDM) applications. According to the test results, incorporating the Virtual Meter may significantly boost the performance of CIM based solutions without jeopardizing CIM compatibility. Ill. 6, bibl. 18, tabl. 2 (in English; abstracts in English and Lithuanian).

S. Vukmirović, A. Erdeljan, I. Lendak, D. Čapko. Virtualiojo matuoklio taikymas bendrojo informacijos modelio galimybėms išplėsti // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2011. – Nr. 1(107). – P. 59–64.

Apžvelgiamas virtualiojo matuoklio taikymas bendrajam informacijos modeliui išplėsti. Virtualusis matuoklis yra matuoklių loginė grupė. Norint padidinti pirminio informacijos apdorojimo našumą gali būti naudojamas virtualusis matuoklis. Modeliui išplėsti galima naudoti papildomas nustatymų klases virtualiuosiuose matuokliuose. Atlikti eksperimentiniai informacijos nuskaitymo iš elektros galios paskirstymo įrangos bandymai. Pagal gautus rezultatus atnaujinus bendrąjį informacijos modelį galima gerokai padidinti funkcijų našumą nepakenkiant bendrojo informacijos modelio suderinamumui. Il. 6, bibl. 18, lent. 2 (anglų kalba; santraukos anglų ir lietuvių k.).