www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

Metrol. Meas. Syst., Vol. XIX (2012), No. 3, pp. 521-530.

PAN POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS

Index 330930, ISSN 0860-8229

www.metrology.pg.gda.pl

M&MS

NETWORKED OBJECT MONITOR – A DISTRIBUTED SYSTEM FOR MONITORING, DIAGNOSTICS AND CONTROL OF COMPLEX INDUSTRIAL FACILITIES

Zdzisław Kowalczuk, Jakub Wszołek

Gdansk University of Technology, Faculty of Electronics, Telecommunications and Information, Department of Decision Systems, Gabriela Narutowicza 11/12, 80-233 Gdansk-Wrzeszcz, Poland (✉*kova@pg.gda.pl*, +48 22 432 7721, *kubawszolek@gmail.com*, +48 22 432 7721)

Abstract

The paper puts forward a method of designing and creating a complete computer system for monitoring and diagnosis of business and industrial facilities, as well as for control purposes. The proposed solution represents a computer-network system being a practical tool for communication, control and management of modern plants and enterprises. The applied concept of communication, based on the Service Oriented Architecture (SOA), makes a new attempt to solve certain performance problems met when using a (previously developed) Networked Object Monitor (NOM). The principal idea of increasing the performance of NOM lies in employing a common data bus, refereed to as a Diagnostic Service Bus (DSB), in the NOM monitor. The paper also describes a preliminary concept of a network description language (SMOL), which is designed to describe the functions, mechanisms, and network devices and to be a basis for simulation and verification of the NOM-monitor function.

Keywords: diagnostics of industrial facilities, diagnostic service bus, network object monitor.

1. Introduction

In recent years, processes of development of the so-called intelligent buildings, both in Poland and abroad, are very dynamic. This applies to the single-family housing sector as well as to industrial facilities, where substantial development changes are also noticeable. Investing in modern high-tech concepts of border automation, electronics and information technology not only creates a financial benefit bound to the economy of operating objects, but also contributes to the protection and care of the nature that surrounds us.

Considering the case of smart or 'intelligent' objects, there are two main branches, which contribute to the development of technologies for such 'intelligent' products as single-family housing and industrial facilities. In both of them, part of the available automation solutions is common and performs a similar job. From a practical viewpoint of such projects, quality and reliability of the applied equipment are most important. With growing interests in 'intelligent buildings', and a dynamic development of companies implementing such modern solutions, the availability and openness of the solutions has good reasons. Along with the development and implementation of innovative ways to manage intelligent buildings, many scientists are still looking for up-to-date and optimal methods of monitoring, control, diagnostics and integration of computer-controlled systems.

The basic idea of integrated building automation systems was born in early 1970s. First systems were attempting to manage services such as power, lightening, and heating, and were referred to as BAS[1] and EMS[2]. Over the years there were many different concepts of new

[1] BAS – building automation systems

Article history: received on Feb. 7, 2012; received in revised form on June 8, 2012; accepted on June 25, 2012; available online on Sept. 28, 2012; DOI: 10.2478/v10178-012-0045-4.

devices, communication protocols, integration methods that contributed to a new technology branch called BMS[3]. Nowadays the BMS integrates many different systems like power networks, HVAC[4], security, magnetic card and access control, fire alarm, elevators and others. Worldwide companies (like Siemens, Honeywell, Delta Controls and others) still work on this type of dedicated solutions. From the perspective of the user, proprietary solutions are nevertheless closed to products/devices of others manufactures. This becomes a source of practical problems during system integration. Some of the solutions use only one communication protocol, which inevitably limits the range of available devices. Our goal is to propose a system that will be open and will support the greatest number of protocols (such as BacNET, KNX, X-15, LonWorks, EIB), and that, in such a way, will be compatible in terms of communication with multiple devices from different vendors. We also propose to apply the concept of Service Oriented Architecture, which is new in the BMS environment, as well as a simulator useful for designing and implementing of the system.

Networked Object Monitor (NOM) is the result of a challenge to create hardware and software applications that allow not only control and measurement but also diagnostics and event-management in distributed facilities networks [1]. This application is being developed at Gdańsk University of Technology (GUT) since 2009, and gradually increases the scope of its implemented functionality. In this article we present the range of the work done and plans related to the future development of the project.

2. Networked Object Monitor (NOM)

This section relates to the basic system assumptions as well as to the achieved prototyping implementations of the system for distributed monitoring, control and diagnostics performed for the so-called intelligent objects (plants or buildings).

2.1. Basic assumptions

The main aim of this work is to develop a fully autonomous standalone solution. It should allow us to monitor and control intelligent buildings and networks of interconnected objects, in real time. The multitasking and flexibility of the developed system should allow for using it for handling not only single-family houses but also different industrial facilities.

The first phase of this project was associated with the development of a relatively simple mechanism to transfer information between transceivers (transmitters and receivers), which enable the construction of multi-layer wireless networks [2]. Selection of appropriate technologies and development of system architecture and implementation allowed us to obtain a solution enabling open transfer of information and control of all devices connected to the network nodes. The proposed solution assumes the construction of a star topology network, managed by one central system, called the co-ordinator of the network. In our prototype design of the wireless network the industry standard called ZigBee[5] was applied to messaging (i.e. for sending messages). In subsequent studies we applied also the WiFi[6], Ethernet[7], RS-485 to extend the possibilities of communication between diverse network modules needed by the end user.

[2] EMS – energy management systems
[3] BMS – building management systems
[4] HVAC – heating, ventilation and air conditioning
[5] ZigBee – dedicated robust high-level communication protocol using digital radios based on IEEE 802 standard.
[6] WiFi – communication standard that allows electronic devices to exchange data wirelessly over a computer network.
[7] Ethernet – standard for exchanging data in computer networks (IEEE 802.3).

Metrol. Meas. Syst., Vol. XIX (2012), No. 3, pp. 521-530.

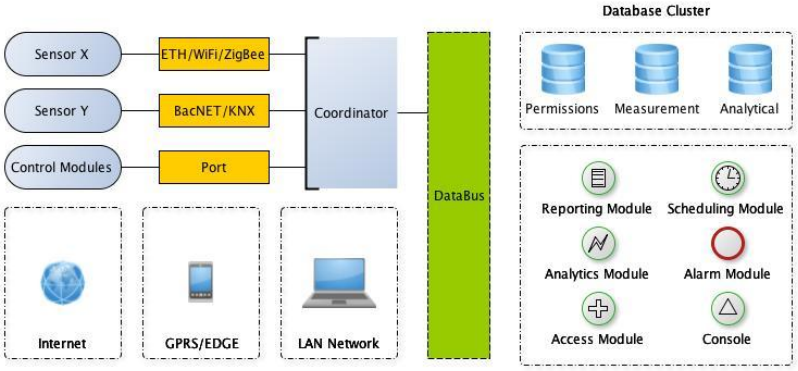

Fig. 1. Structure of the Network Object Monitor.

The system employs a scheme based on the assumption of retrieving information from sensors and then transmitting this information to the network coordinator, which ultimately directs it to a declared computer system for further processing.

2.2. Hardware layer of NOM

The coordinating module is a primary element of the hardware layer, responsible for tasks associated with receiving, processing and transmitting information incoming from sensors. The architecture of this module is based on the ARM[8] (AT91RM9200) 32-bit micro-controller. Using this IC system (a specific integrated circuit) enables the use of its large computing capabilities with relatively low energy consumption. We equipped the micro-controller module with a dedicated operating system, which was developed using a free operating system – Linux[9].

Based on the capabilities of the operating system there were created several dedicated software tools that allow not only the coordination of the wireless network activities, but also the immediate visualization of how the network operates. The user can also be informed about different problems and anomalies that occur during communication between the measurement gauges and the master device. The master module communicates with the sensors via wireless technology. The presented project was implemented with the use of ZigBee, WiFi and GPRS[10] communication standards. When choosing a method of communication the system designer can then focus solely on the constraints imposed on the system. It is also possible in our solution to use several communication mechanisms implemented at the same time.

In addition to the device for managing and coordinating the network, there is also a part responsible for making the measurements and their transmission, as well as for generating feedback in the form of control signals. The modules responsible for measurement and control are referred to as endpoint modules. The performance of the application usually relies on measurements carried out by the measuring units (endpoint modules), from which the information is sent to the coordinating module (coordinator). Note that precise analysis of the measurement information can also be made there. In the first stage a record of information is made to the database, what can be followed by processing that information (e.g. averaging) with the use of past samples. It is clear that using a kind of backward approximation allows us to avoid unnecessary changes to the control system for single false readings of the measuring system. Important are also (and we have considered them) test situations in which incorrect information is propagated in a number of consecutive samples. If we take into account

[8] ARM – advanced 32-bit microcomputer based on RISC (Reduced Instruction Set Computer) architecture.
[9] Linux – open-source computer operating system.
[10] GPRS – Global Packet Radio Service.

erroneous measurements that go beyond some definite extremes, then we can use a specific constraint that makes a basis to inform the managing person about erroneous (false) readings detected in the system.

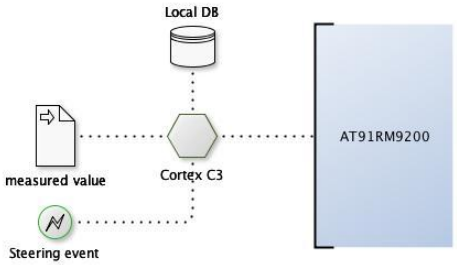

Fig. 2. Connecting a measurement endpoint device to the coordinator.

Prototypes of the measuring and controlling modules were built with the use of microcomputers ATmega32[11] (very popular on the market), which – in the next system edition – were substituted by Cortex M3[12] microcomputers.

2.3. NOM – software layer

The software part of the application is divided into several functional modules corresponding to the implemented hardware. Dedicated software solutions were created for each of the modules described above.

The coordination module implemented is based on maintenance services (unix *daemons*), written with the use of the available functions of the operating system. The services allow us to refer to external devices via the API[13] system. The architecture of the application takes into account the possibility of communication between system services using the network connections. The proposed solution enables optimal allocation of tasks into several smaller applications, what contributes to the efficiency of implemented solutions, and provides greater reliability ('stability') of the system in case of failures. Note that network reliability characterizes computer networks in terms of data transmission medium, which should be robust to disturbances, and – to a certain extent – to changes of the network load (*i.e.* the level of the generated traffic).

The network coordinator has separate services responsible for monitoring and processing of measurement data from sensors that communicate in a given broadcast standard (*i.e.* ZigBee, WiFi, and GPRS). We also apply a supervisory service, which is responsible for the ongoing validation of the other modules. In case of failure in one of the working services, the supervisory service tries to reset it and automatically notify the maintenance team, or the person supervising the work of the system, using text messages or emails.

Measurement data and rules describing (under definite conditions) the reaction/behavior of the system are stored in a dedicated file-based database. An external database server (with the ODBC[14] connection) is also designed for the system user. This saves space and increases computing capabilities by embedding the computing applications on the database server in the form of stored procedures [3].

Measurement and control modules are designed specifically to take advantage of relatively small, low-voltage microprocessor chips. In the first stage of the project, assuming only one

[11] ATmega32 – microcontroller desgined by Atmel Co, equipped with a 32kB flash memory.
[12] Cortex M3 – 32-bit microprocessor for highly deterministic high-performance platforms.
[13] API – Application Programming Interface.
[14] ODBC – Open Data-Base Connectivity.

Metrol. Meas. Syst., Vol. XIX (2012), No. 3, pp. 521-530.

method of data transmission (ZigBee), we employed Atmel micro-controllers, model AVR[15] (having such a transmission module).

Currently, we are in the phase of designing and implementing new test modules based on a much more efficient computing unit, Cortex M3. Using more powerful processors, we are able to solve a number of performance issues that arise during operation of these modules in complex and noisy environments. With such facilities of the existing implementation, some of the tasks associated with digital filtering of measurement signals can be performed right away at the level of the measurement and control endpoint devices. This allows us to design distributed systems of data processing and, in consequence, to send already pre-processed data to the coordinator. The endpoint modules are managed by the open-source operating system FreeRTOS[16]. As has been observed during tests of the NOM system in monitoring intelligent buildings, quite often the system appears in states in which only part of the endpoint modules is computationally busy, while the remaining modules are not sufficiently exploited. Therefore, at the present stage of the design we are considering the possibility of creating an efficient distributed computational environment which should enable transfer of current computational tasks between the endpoint modules.

The NOM application software layer provides a number of functional tools for diagnosis and management of monitored objects. One of them allows for visualization of measurement values and generating plots, as well as for archiving current data for future analysis. The backend part of this tool was implemented in PHP[17], whereas a user interface part responsible for displaying plots was programmed based on a JavaScript library called FLOT[18]. Another tool, created in the same technology, is dedicated for managing the whole platform and its devices. The user is not only able to monitor the current state and behavior of the system, but also has all the necessary possibilities for interaction. Through this application one can control devices directly connected to the endpoints of the NOM application. Moreover, user interface functionalities were created with the use of asynchronous JavaScript and XML solutions [4].

Currently, we work on a tool for monitoring the quality of transmission between the central coordination module and the endpoint devices. What is important, most of theses applications are accessible from web browsers. Such a solution allows the users to remotely access the monitored object from anywhere in the world, without having to install any specific additional software. In the process of development of the project, a number of various dedicated tools for testing and diagnosis of the composed structure of the monitoring network NOM under construction were designed. Most of these tools are still in development and will become part of the self-diagnostics package in the near future. We also develop a user-friendly programming interface that allows the users to build their own NOM structure. Such a solution makes it possible for the persons who monitor the performance of intelligent buildings, to build their own concepts of diagnosis and control.

2.4. Integration of hardware and software layers

When developing complex systems the designer often encounters the problem of integration of the hardware layer with dedicated software. On the same basis, in the analyzed case of NOM, a number of implementation issues had appeared, directly affecting the reliability of the network being implemented.

One of the most interesting challenges of integrating the capabilities of hardware and software concerns the functionality associated with the intended distributed processing. In our

[15] AVR – developed by Atmel, modified Harvard architecture of the RISC single chip micro-controller.
[16] FreeRTOS – real-time operating system for embedded devices.
[17] PHP – server side scripting language, http://www.php.net .
[18] FLOT – JavaScript plotting library of OpenSource, http://code.google.com/p/flot/.

case, an appropriate procedure is also necessary to optimize the system performance and to comply with (real-) time requirements.

It is also important to adequately coordinate the paths of transmitted messages, as every time when the message is bouncing between the nodes it needs to be handled by a message management system. From the system and user point of view it is essential that information (*i.e.* measurement), which requires digital processing, issued from an endpoint module A, after processing by a module B, goes to a defined destination C (i.e. the network coordinator). This raises a number of problems related to the correctness of transferred data, encoding and decoding. In advanced network solutions, the ability to prioritize the transmitted information packages is also very important, allowing the user to define the validity of transmitted and processed data. It is thus apparent that working on the NOM project constantly brings new challenges and opportunities for integration of its interdependent layers.

2.5. The NOM simulator

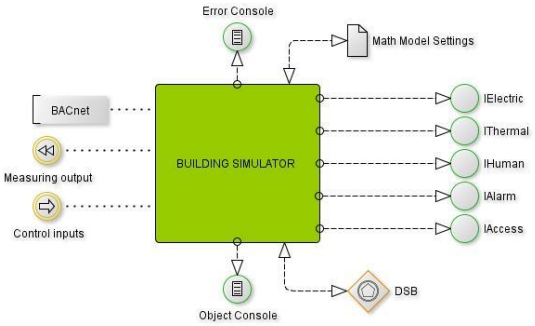

Fig. 3. General configuration of the simulator of an intelligent building.

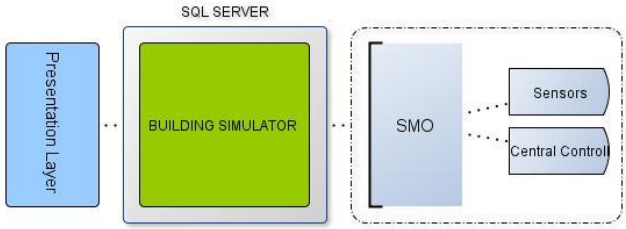

Fig. 4. Architecture concept of the simulator of an intelligent building.

In the course of system development, the need to create conditions similar to real ones contributed to the development of tools for carrying out simulations of the NOM system working in a definite environment [5, 6].

Our current work concentrates on adaptation of the NOM platform to work in the so called ‘intelligent buildings’, which should be ‘green’ – in response to the European Union (i.e. should implement the ecological recommendations of EU [7]). In this context, in particular, the task of the simulator is both to generate test signals for the NOM system and to response to varying control signals sent from the NOM machine. The simulator is a perfect solution for the analysis of operation of NOM in strictly-defined configurations [5, 6]. A general concept of simulation [5] is presented in Fig. 3.

The concept of an intelligent building simulator means a computer application which can simulate responses of monitoring objects in a certain ‘real’ environment. Currently, the simulator provides calculations based on the Oracle[19] database engine. In the prototype of this application we made a distinction between an application logic layer and a database layer. In general, it appears to be a good practice from the computer architecture viewpoint. During the development process, however, we concluded that the performance of computations was much higher when using directly a database engine.

This observation contributed to system migration from the typical model of application (separate application data and logics layers) to the integral solution that stores both data and logics directly in a database. Thus the logics part of the application were (re-)written to

[19] Oracle - http://www.oracle.com/us/products/index.html .

Metrol. Meas. Syst., Vol. XIX (2012), No. 3, pp. 521-530.

definite stored procedures of the PL/SQL[20] database language. Then the respective compiled application has only to call such a procedure (instead of connecting to and downloading data from the database). This solution extended the system performance and accelerated the calculations over 20 times. A general concept of the simulator is presented in Fig. 4.

3. DSB – Diagnostic Service Bus

Basic assumptions and principles of a dedicated bus for data exchange in the NOM systems of networked object monitoring are presented below.

3.1. Assumptions

During the test work related to simulation of failures, which may occur in the NOM system, the most critical point in the network turns out to be the coordinating module. Any failure to this supervising module makes the entire diagnostics and control system stop working properly. We decided to solve this problem using the SOA[21] architecture model. It is a well-known concept used to build information systems, focused on providing services [8].

With the use of the concept of SOA we are able to design a totally new model of the system architecture [9] based directly on the common services-oriented data bus, called the Diagnostic Service Bus (DSB).

In such a way, implementing a completely new communication medium, we sort out the problem of a weak point (represented by the network coordinator) in network structures with a star topology. Certainly, the new architecture still uses a common data bus, a universal medium that allows for the connection of individual modules of measurement and control, data processing applications, database systems, as well as supervisory functions.

The basic architectural concept is based on delivering measurement data through a common data bus. Any measurement endpoint device can create a direct connection through a communication channel to the DSB. The bus has suitably implemented routing mechanisms, which allow creating definitions of data routes. Such a definition describes every single step of passing (measurement) data. The steps are elementary parts of the flow that represents the whole measurement-data route required. A testing scenario is presented in Fig. 5. It describes passing of data from a measurement device through Pre- and Post-Transformation modules (that create messages in a proper format), as well as the service which transforms messages, performs additional calculations and passes data through the inference module. The service can also address the processed data through the bus to another service or place them in the database. The methods of data transfer depend on the flow configuration created by the end user.

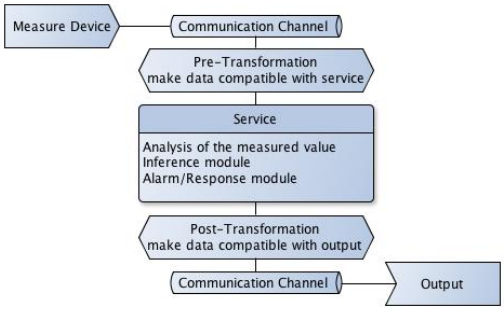

Fig. 5. Diagnostic Service Bus – basic architecture concept.

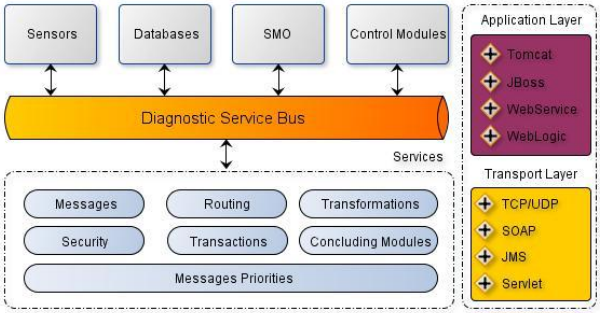

Fig. 6. Diagnostic Service Bus.

[20] PL/SQL - Procedural Language/Structured Query Language typical for Oracle.
[21] SOA – Service Oriented Architecture, a method for designing and development of software.

During the development of the DSB, the modes of action and the methods of implementation of other service-oriented systems, such as MULE[22] [10] and Biztalk[23] [11], were also analyzed. The experience gained was pivotal in implementing solutions tailored to the needs of the NOM application.

3.2. Networked Object Monitor (NOM) and Diagnostic Service Bus (DSB)

Integration of the developed concept of the NOM platform with the designed data bus, Diagnostic Service Bus (DSB), however, requires major changes in the architecture of the original NOM system. Namely, the applications previously constructed had to be rewritten or in large part to be at least modified.

One of the basic assumptions upholding the design of the DSB was the opportunity (option) to connect to it devices and applications that had implemented support for the TCP/IP[24] protocol. In this way, from the perspectives of such complete measurement and control systems, we are able to create solutions that facilitate the integration of measurement and control modules with diverse systems for information processing and data interfacing. Using the applied DSB solution, the format of the messages sent is compatible with XML[25].

The method of operation of the NOM monitor is yet based on collecting information by the measurement and control modules, converting the respective signals to the ruling standard (pre-processing), and transferring the data to the DSB bus. At the next stage of data transfer, the usefulness of the information collected is assessed, and its interaction with other data is performed in an inference system.

A great advantage of a common data bus consists in the possibility of concurrent processing of the information available on the bus [12, 13]. The data can then be processed at the same time by several processes (applications) which use DSB simultaneously (in parallel). It is possible, for example, to make visualization and processing the same information transported, at the same time.

From the standpoint of managing messages on the DSB bus there is a possibility of total control of the direction and method of data transfer and assigning certain priorities to the information transmitted.

The diagram of operation of the Diagnostic Service Bus is depicted in Fig. 6.

Functionality, which is currently under study, is related to compliance with the standards posed on real-time systems, because the DSB bus tested at this stage of development – having a large amount of processed information – has not met these requirements.

Currently, the SCADA[26] system is widely exploited in automation as a standard computer system used for monitoring technological or manufacturing processes. This system allows you to integrate the PLC[27] controllers into one network. The PLC devices transmit data (over the network) to a computer system, where they are subsequently archived, processed and subjected to visualization.

Thus, in the above sense, the development of the NOM system is our response to the standard SCADA systems commonly used in industry. By designing the NOM-DSB system, we hope that it will be able to compete with the SCADA systems as a universal tool in constructing management and control systems of various intelligent objects (and buildings).

[22] MULE – open-source lightweight enterprise service bus.
[23] Biztalk – commercial realization of the enterprise service bus based on the SOA model.
[24] TCP/IP – computer network protocol; precisely, Transition Control Protocol/Internet Protocol.
[25] XML – Extensible Markup Language.
[26] SCADA – system for supervision/monitoring and control of industrial infrastructures or production processes.
[27] PLC – Programmable Logic Controller.

Metrol. Meas. Syst., Vol. XIX (2012), No. 3, pp. 521-530.

4. SMOL – network description language

As the next stage of development of the Networked Object Monitor, we chose to develop a dedicated language (SMOL) for description of features, mechanisms and networked devices. The semantics of this language is to enable precisely mapping of the relationships existing in real network systems. The scope of the work undertaken will thus include a parser that allows translation of the codes written in SMOL into a simple language understood by the computer environment. The proposed development of an environment for the SMOL language is to enable the use of two component libraries: libraries for precise mapping of the network structure and a library containing a collection of the features of the Networked Object Monitor. The respective task of the system designer will then be to define requirements (both structural and functional) of the designed system.

The resulting parsing process will allow to generate graphics of the network structure and to determine the necessary systemic conditions, taking into account structural optimization considering the posed constraints.

The intended language is to be integrated with the previously developed simulation environment, which will be used to run tests to verify proper operation of the designed NOM system in the aspects of monitoring, diagnosis and control.

For the grammar analysis we utilized ANTLR28, an open source tool which provides a framework for constructing recognisers, interpreters and compilers.

The listing given in Appendix demonstrates an exemplary application of the concept of the SMOL programming language under design for the description of a small piece of a NOM network. As the result of the analysis performed by the SMOL parser, an image describing connections between the nodes of this exemplary NOM system is generated, as shown in Fig. 7.

Having the system described in the SMOL language and verified by the parser both syntactically and semantically, as well as graphically (in the form of a network structure), the designer will be able to carry out effective performance/load testing in the associated NOM simulator.

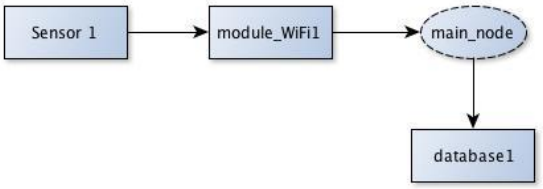

Fig. 7. Example of a SMOL parser output.

5. Conclusions

The concept of using computer systems for diagnosis, monitoring and control is not new. For example, the SCADA system has long been in industrial use. However, contemporary very rapid development of the frontier fields of automation and information technology makes it possible to design new, far more efficient computer network systems having a wide range of built-in functions and features demanded by the end user.

The proposed project of a contemporary monitoring system of networked objects (referred to as NOM) is an attempt to create an application that fulfils the ever-increasing demand for solutions dedicated to monitoring, diagnostics and control of a single object, or a composed network of industrial facilities.

[28] ANTLR - http://www.antlr.org/ .

In the next stage of development of NOM, we are developing a dedicated description language SMOL, suitable for describing features, mechanisms and devices of such computer network systems. The semantics of this language is to enable precise mapping of all the necessary relationships of real network systems.

References

[1] Kowalczuk, Z., Wszołek, J. (2009). *Network monitoring and diagnostics of buildings*. In [12], 227-234. (in Polish)

[2] Kowalczuk, Z., Wszołek, J. (2009). Monitoring objects over networks. *Systems Science*, 35(3), 49-53.

[3] Feuerstein, S., Pribyl, B. (1995). *Oracle PL/SQL Progamming*. O'Reilly Media Inc., California.

[4] Eckel, B. (2006). *Thinking in Java*. President, MindView, Inc., Massachusetts.

[5] Czarnecki, P., Dopka, M. (2010). Sub-optimal management of heat processes in intelligent residential buildings. *Raport ZK104i. Dept. of Decision Systems*, Faculty of ETI, Gdańsk University of Technology. (in Polish)

[6] Mydłowski, M., Pisz, B. (2010). HVAC Philosophy in intelligent office building. *Raport ZK104i. Dept. of Decision Systems*, Faculty of ETI, Gdańsk University of Technology. (in Polish)

[7] Kowalczuk, Z. (2010). *Energy Performance in Buildings.* PWNT, Gdańsk. (in Polish)

[8] Chappell, D. (2004). *Enterprise Service Bus: Theory in Practice.*

[9] Erl, T. (2009). *Service-Oriented Architecture*. Pearson Education, Inc. Indiana.

[10] MULE (January 2012). *http://mulesoft.org/*.

[11] BIZTALK (2012). *http://www.microsoft.com/biztalk/en/us/default.aspx*.

[12] Kowalczuk, Z. (2009). *Detecting, Analysing and Fault-Tolerant Systems.* PWNT, Gdańsk. (in Polish)

[13] Kowalczuk, Z. (2009). *Diagnosis of Processes and Systems*. PWNT, Gdańsk.

Appendix.

Exemplary application of the SMOL programming language for the description of the NOM network from Fig. 7.

```
#Use of external libraries
use SMOF;
use SMOH;

#Elements definitions
define sensor1;
define modul_WiFi1;
define main_node;
define database1;

#Inheritance
sensor1 :: sensor;
module_WiFi1 :: WiFi;
main_node :: main;
database1 :: database;

#Parameter settings
sensor1 -> name("Temperature sensor");
sensor1 -> unit("Celsius");
module_WiFi1 -> name("WiFi
communication module");
module_WiFi1 -> standard("N");

database1 -> type ("Oracle");

#Connections definitions
sensor1 -> connect(module_WiFi1);
module_WiFi1 -> connect(main_node);
database1 -> connect(main_node);
```