DiaSter - Intelligent System for Diagnostics and Automatic Control Support of Industrial Processes

Submitted 27th June 2011; accepted 1st September.

Michał Syfert, Paweł Wnuk, Jan Maciej Kościelny

Abstract:

The paper presents general description of the DiaSter system implementing advanced methods of modeling, diagnostics and supervisory control for industrial processes. The scope of the tasks realized in the system as well as the system software platform were characterized, in particular: the software structure, central archival and configuration databases, the way of data exchange in the system and the modules of modeling and calculations.

Keywords: *diagnostic and supervisory systems, automatic control support, software platform.*

1. Introduction

In recent years there have been significant developments in techniques for modeling [3], [5] advanced control [1], [11] and process diagnostic [2], [4], [6], [10]. Modern computer systems enable the application of complex computational algorithms, developed on the basis of recent re-search in computing, automatic, diagnostics and know-ledge engineering. They use artificial intelligence techniques such as artificial networks, fuzzy logic, rough sets, evolutionary algorithms and methods for knowledge discovery from databases [9].

Development of methods outlined above in conjunction with the rapid progress of computer technology (computing power of new generation processors, memory capacity, speed of data transmission in LAN networks and field bus, the Internet growth) leads to a new generation of control systems. It is characterized by the introduction of advanced software for modeling, control and diagnosis processes. This software is a special software modules, which are part of the automation system, or expert systems integrated with automation systems.

Such software package is DiaSter system. It is developed by a research team composed of specialists from the Warsaw University of Technology, Silesian University of Technology, Rzeszów University of Technology and University of Zielona Gora and supported by polish grant: An intelligent system for diagnosis and control of industrial processes support DiaSter. It is brand new, functionally and software extended version of AMandD system, developed in Institute of Automatic Control and Robotics, Warsaw University of Technology [7], [8]. The system is dedicated for use in the energy industry, chemical, pharmaceutical, metallurgical, food and many other.

The system is world-wide unique solution. It includes implementation of a wide range of the latest algorithms in the field of intelligent computation, used to system modeling, supervisory control, optimization, fault detection and isolation. Thanks to its open architecture, connections to virtually any automation system are possible and easy to implement. The position of DiaSter system in industrial process management tasks is presented in Fig. 1.

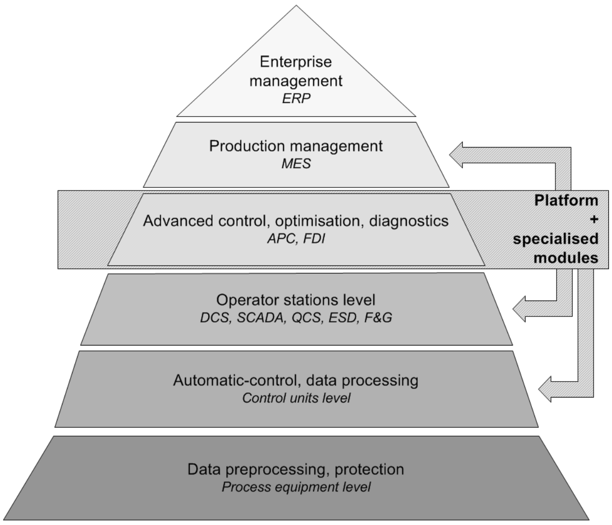

Fig. 1. The position of DiaSter system in respect to the hierarchy of tasks (and system classes) connected with production process.

2. DiaSter system functions

DiaSter system allows to realize following functions:

- **Process variables processing.** System gives a possibility to freely design processing paths for each variable.
- **Simulation and modeling.** In the system is possible to create models on the basis of real physical description of modeled phenomena, or to identify model on the basis of measured data.
- **Virtual sensors and analyzers.** Virtual sensors based on the analytical, neural-network or fuzzy models can be treated as information redundancy for real measurements.
- **Process simulators.** Developed within DiaSter system process models can be used as simulators for operator training, or to test new strategies of control or to optimize the process set-points.
- **Fault detection.** In the system methods based on analytical, neural-network or fuzzy models as well as heuristic tests utilizing different kinds of relations between process variables are used for fault detection. Such detection methods give a possibility for early detect of much higher number of faults than classic alarm system.

- **Fault diagnosis.** In DiaSter system two independent diagnostic inference mechanisms are implemented. First one allows fault isolation based on the analysis of the set of actual fuzzy diagnostic signal values and the relation between faults and symptoms stored in the base of knowledge. Second uses belief networks and multi-facet models.
- **Monitoring of degradation degree of technological equipment.** The system allows to early detection and tracking of slowly developing destructive changes. In addition an approximation of time left to critical state can be done.
- **Support of process operators decisions.** On the basis of elaborated diagnosis the DiaSter system can additionally support the process operators decisions in abnormal and emergency states.
- **Knowledge discovery in databases.** There are mechanisms for knowledge discovery in SCADA and DCS databases implemented in system, designed mainly to support diagnostic reasoning.
- **Advanced control and optimization.** The algorithms of superior predictive control (DMC, GPC) using linear models, as well as algorithms based on nonlinear models, in particular fuzzy and neural networks based, are implemented in the DiaSter system.
- **Superior tuning and adaptation of control loops.** DiaSter is able to carry out both preliminary (pre-tune) and precise (fine) tuning of control loops with step response or frequency methods.

3. System structure

DiaSter system consists of **a software platform and several specialized packages** cooperating with its use.

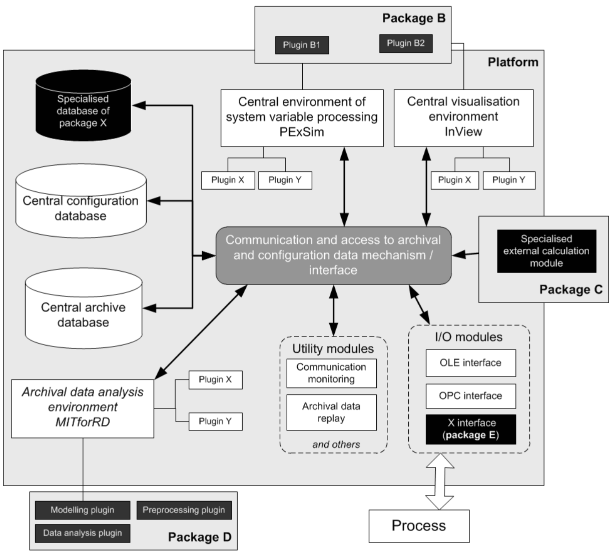

Fig. 2. General structure of DiaSter system. Black blocks symbolize specialized packages realized as different system components cooperating with the platform.

The software platform is a core of a system. Its main components are: archival data processing and model identification module, module of on-line processing of system variables, visualization module, central configuration and archival databases and communication server. They are described in the consecutive sections.

There are also available several utility module. There are not responsible for main information processing but are very useful during system configuration and tests. Good examples are communication scanner monitoring messages exchanged between calculation modules or module reconstructing archival process variables from database (or files) in real-time, possibly with some acceleration factor.

Advanced system functionality is delivered, developed and realized by specialized packages, called user packages. They are realized as independent software modules cooperating with other system components or in the form of plug-ins of software platform modules (modeling, processing and visualization).

4. Central archival and configuration databases

In order to make possible the cooperation of all system components the following elements were worked out: common information model, central configuration environment and central archival database.

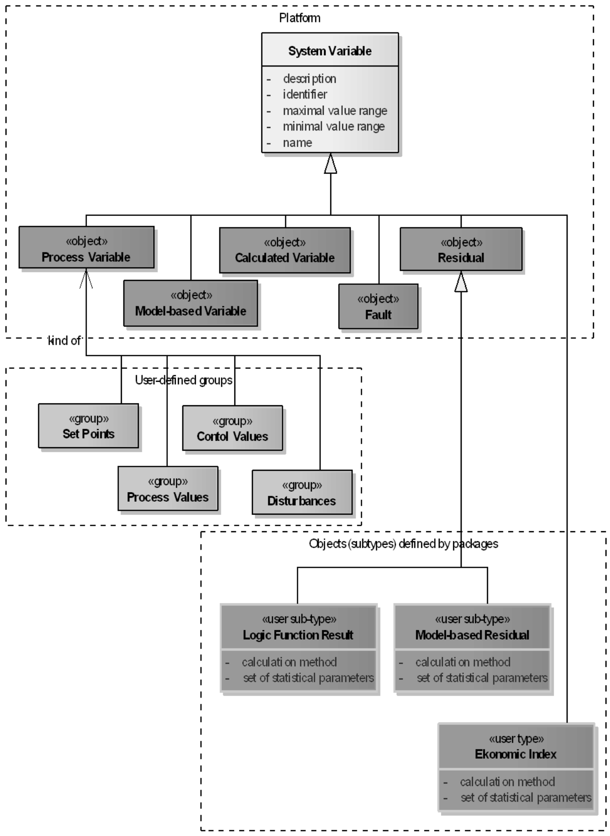

Fig. 3. Main objects and relations of the information model.

The **information model** specifies the definition of the information being exchanged between the system components (Fig. 3). It constitutes the basis of the data exchange in respect to configuration as well as processed system variables. First of all, the information model defines:

- the elements connected with process/installation description, e.g., division into subsystems and particular components including the specification of logical as well as physical units, specification of process variables and their groups,

- the logical components of processing algorithms connected with realized by the system tasks, e.g., models, residuals, calculated and simulated variables, faults,
- the relation between the above elements, e.g., the relation "is a part of" and "is connected as input / output" defined between process subsystems and components or the relation "controls the behavior of" between residuals and monitored process components.

The information model also defines the types of system variables and the types of their values processed by the platform. It constitutes the framework of process and processed variables description. It is designed in such a way to be extendable by the particular user packages, or a group of packages. In this area the platform enables:
- creation (registering in the platform) user-defined data types, e.g. fuzzy signals,
- creation (registering in the platform) user-defined system variables types, e.g., efficiency indexes,
- the division of system variables into user-defined groups, e.g. control (CV), set-point (SP) and controlled (PV) and disturbance (D) signals,
- creating user-defined types of the process components, e.g. pipelines, valves,
- the possibility to register user-defined relations between different system elements.

The **central configuration environment** is a repository of configuration data for all system modules. It consists of:
- Central configuration database. It is a relational database that stores the whole system configuration for particular application.
- The set of configuration interfaces. They are used by all system modules and their components (including plug-ins) to get standardized access to the configuration data. They delivers the set of necessary and useful functions for manipulating the elements of system configuration.
- The interface to user-defined databases. This interface enables to create, manage and access specialized databases assigned to system packages, e.g. the database of cases used by case-based reasoning methods (CBR) or diagnostic messages database. Such databases can be used to store any data (configuration or variables) typical for particular package, or a set of packages. The data stored in those databases do not have to be consistent with information model of the system platform.
- Central configuration module. It is used to manage common for all system modules configuration data that is consist with information model.

The **central archival database** is capable to store any system variables defined in the system configuration. It can store the variables of build-in data types (process variables, residuals, faults, etc.) as well as user-defined data types. Also any type of variable values can be stored. There build-in as well as user-defined value types are handled.

The variable values can be stored in short- or long-term archives. Each of the system component (modules, plug-ins, build-in and user-defined) has access to specialized interface constituting common access to archival data.

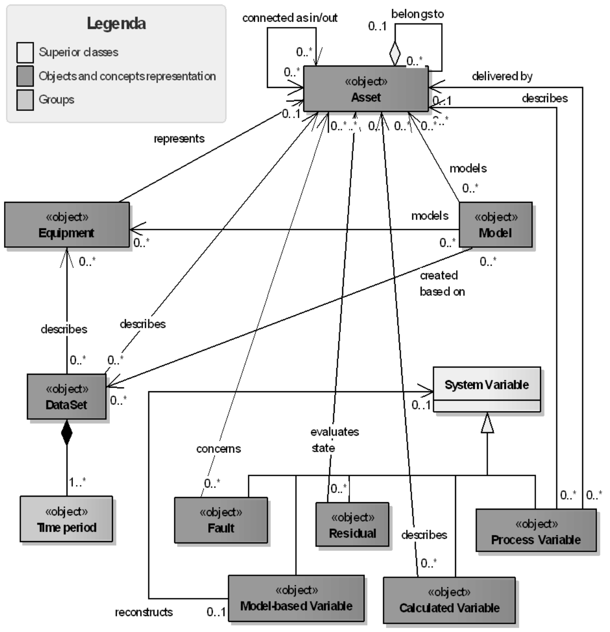

Fig. 4. Example of extending platform information module by adding user-defined variable types and groups.

5. Data exchange

DiaSter system is fully distributed software system with possibility to work on many PC-s connected with local area TCP/IP compliant network. Communication subsystem is a native solution, working with specially designed protocol independent on any external software. Communication library delivers simple API to system modules programmers. Thanks to this library the connection of an external program to DiaSter system is reduced to loading library and setting IP address of communication server. To send a message only one function call is needed.

Base communication method in DiaSter platform is based on message transmission between modules in two modes: direct module to module, and broadcast to topics (publish/subscribe). In both modes in data exchange active role acts MRIaS server included in the platform. Server is only used to transmit information between modules and to provide transmission errors maintenance. It is not responsible for message analysis or storage. Communication is always made through TCP/IP sockets, regardless of whether the modules work on standalone computer or on multiple networked PCs. Mentioned mechanisms are mainly used to transmit the data rapidly and frequently changing, e.g. the values of measurements, the result of current calculations, etc. Any type of value acceptable by the software platform (floating point numbers, binary signals, fuzzy values, arrays, etc.) can be transmitted this way. Additionally, it is possible to call remote procedures - both in direct communication (module A calls procedure executed by B and receives the results) and broadcasting (e.g. supervisory or configurator module send a "reconfigure" request to all modules within a single RPC call).

More sophisticated communication mechanisms are provided by some modules working on-line. Additional communication interfaces build on the basis of CORBA standard are used to access the properties and methods of each function block embedded in the on-line calculation module (PExSim), and to remotely control the process of simulation / calculations realized by this module. Similar mechanism is used in visualization. It allows to transmit complex and infrequently changing information and direct access to system objects.

6. Modeling module

For the identification purposes the **MITforRD (Model Identification Tool for Diagnosis and Reconstruction) module** is designed in DiaSter system. It allows to create models without the knowledge of the analytical form of the relationship between modeled variables. MITforRD module allows to identify both static and dynamic models of different types starting from well-known linear trans-mittances to neural network or fuzzy logic based models. The identification is carried out off-line using measure-ment data from central system archives. The models im-plemented in MITforRD module belongs to the group of partial parametric models of the process variables (time series). Models are obtained in a semi-automatic identi-fication process. The deep knowledge of processes and their physical characteristics is not necessary – module effectively supports users not familiar with identification techniques during the whole identification process. The main features of the software includes:

- common interface to many kinds of models,
- wizards with default parameters for each identification step
- flexible, self-configurable distributed calculation envi-ronment allows the use of free computing power of offices PCs,
- plug-in based architecture allows to easily extend module functionality by the independent software vendors.

MITforRD supports the user in each identification step, starting at data acquisition, up to final analysis and verification of received model. All the time user can access to program options with simple and intuitive menu.

During process data analysis MITforRD allows to edit process archives inside advanced embedded editor and data visualization. Additional features are available *via* attached plug-ins: import and export of data, histograms, statistical parameters, frequency-domain analysis, corre-lations, transformation according to given mathematical expression, filtering, numerical differentiation and integration. An extra functionality is provided to work with process archives filling-in missing samples, validation of signals. Missing features can be added by the user by creating a new plug-in.

One of the main assumptions for DiaSter system was to make it flexible and expandable. Therefore, the main program MITforRD Model Builder does not implement a possibility of model estimation of any type. All model types are implemented as plug-ins. Similar mechanism is provided also in other DiaSter modules. All plugins are based on dynamic link libraries (dll's). In the basic speci-fication of dll libraries only the export of functions is allowed. Thus, in whole DiaSter system additional layer is added, that allows to work with objects. This approach allows to create new plug-in in different programming languages (most commonly used is C++).

MITforRD module may employ a wide range of evolu-tionary algorithms in order to explore the structure of linear or fuzzy models. Such algorithms, besides many advantages, also have one serious disadvantage – usually are very time consuming. To reduce computation time MITforRD provides distributed computing environment. It can be used to utilize free computation power of the classic of?ce PCs running on Windows and connected to the local area network. The environment does not require any change in the local PCs configuration and does not disturb its normal operation. The main PC with running MITforRD Model Builder is a control center for the calcu-lations. There is a built-in calculation manager inside the Model Builder capable to distribute the calculations among local host as well as remote PCs available at the moment. The computation environment built-in into MITforRD module do not require configuration phase, and do not add time overhead caused by technologies like COM, CORBA, etc.

7. On-line calculation module

The main element of DiaSter platform for on-line use is an **calculation module called PExSim**. It is dedicated for advanced system variables processing. The processing al-gorithms are written and stored in the form of configurable function block diagrams. The primary task for PExSim module is to process the information circulating in the system in a way defined by the user. From this point of view PExSim module can be treated as specialized pro-gramming language. An algorithm of information pro-cessing is defined graphically by creating so-called pro-cessing paths. Each path consists of a set of interconnected function blocks, which carries out various tasks on signals (see Fig. 5).

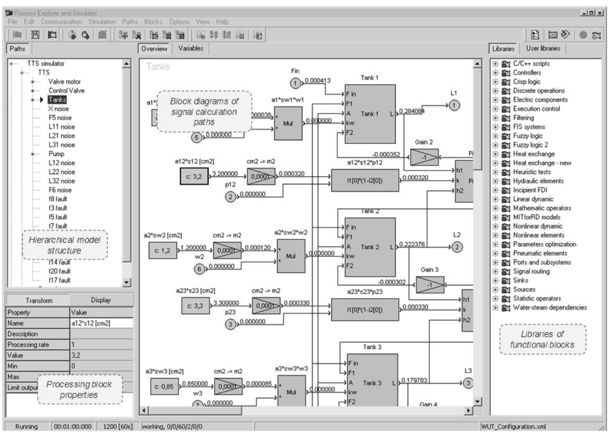

Fig. 5. Graphical programming of desired calculations in a form of processing paths.

Function blocks are provided as PExSim plugins, and are grouped into thematic libraries, e.g.: surces, sinks, statistical operations etc. Each function block has the set of parameters defining the way of signal processing by this block, and stored by the platform. Various block inputs and outputs can transmit data of various types (e.g.: floating numbers, fuzzy values, vectors etc.) including user data types. PExSim can run as stand-alone tool (simulator mode), or as a module working in a distributed system (multi-module mode). Multi-module mode allows to exchange the data with other DiaSter modules thus there is a need for synchronization external data with internal simulation clock. To realize synchronization different modes of simulation triggering are available.

Each main path of the calculation module can be assigned to one of the following groups:

- synchronous paths. In this case the simulation kernel triggers given path with constant, previously defined time period, calling sampling time, or its multiplicity,
- asynchronous paths. These paths are triggered by events generated internally (via specialized block) or coming from outside world (a message, RPC call, etc.).

8. Visualization module

The platform delivers also a visualization module (graphic user interface) called PExSim. It can be used to realized advanced operator interfaces. It is organized in a similar way as tools for configuring process mimics in supervisory, control and data acquisition systems (SCADA) or decentralized control systems (DCS). The set of synoptic screens organized in hierarchical structure is prepared during the configuration stage for particular application. Dedicated displays visualizing the values of system variables in particular way are placed on that mimics.

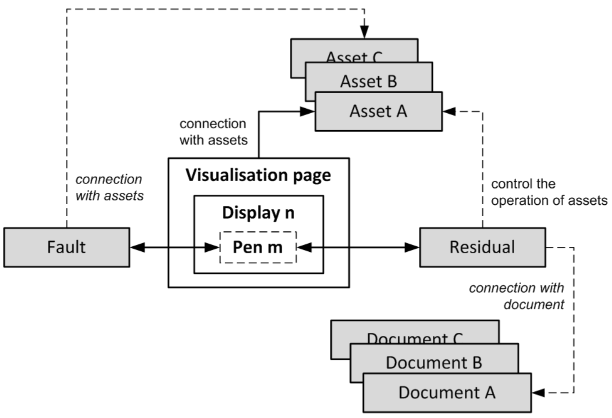

Fig. 6. Integration of user interface with configuration of particular application.

This module is not designed as competitor of well-grounded visualization modules available in SCADA systems. Its main advantages result from its full integration with platform information module. For example, dedicated displays are used to automatically present the information that was used to generate the diagnosis the set of useful residuals in respect to particular fault is retrieved automatically from the diagnostics relation configuration stored in diagnostic package private database. It is possible because the visualization pages can be connected with particular assets, while displays are related with defined faults and residuals. The idea of the integration of user interface components with configuration objects is presented in Fig. 6, while its implementation is shown in Fig. 7.

The displays are realized as visualization module plug-ins. Such approach enables to elaborate specialized displays designed by the users. Such displays can even visualize process variables of user types (unknown by the platform itself), e.g. visualization of fuzzy signals or dedicated diagnostic messages. The only limitation is that the plug-in creator must have the knowledge about the processed signals structure.

The current values of the system variables are automatically delivered by the communication server (including user-defined data types). The module automatically collects the data from central archival database when the history (short- or long-term) is analyzed by the user. Additionally, each display can retrieve the data from the user databases (specialized databases registered by specialized packages started on the platform) with the use of standard SQL interface.

In the case of a need to elaborate very specialized visualization there is a possibility to elaborate stand-alone module. Such module can use the data exchange mechanisms available in the system and cooperate with other platform modules.

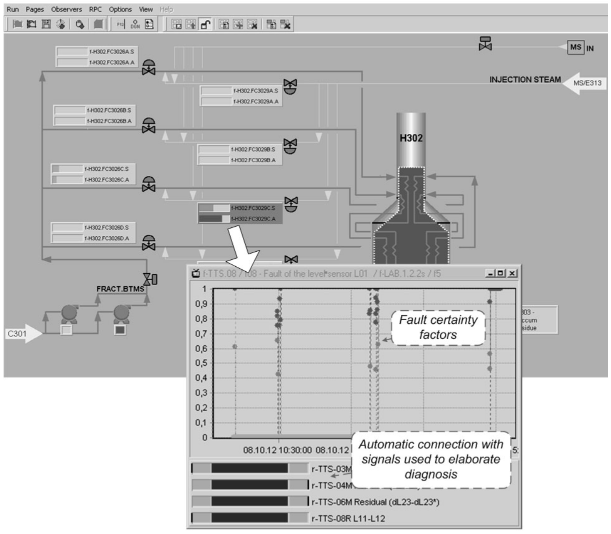

Fig. 7. Exemplary process operator interface dedicated to deliver diagnostics information about process state.

9. Summary

Presented system was created as a result of development grant that finished in 2009. Currently test of the system are conducted. Their aim is to remove bags and finally prepare the system to implementation, commercial ones as well as research-development.

The most important part of the system is its software platform and the possibility to easy extend its functionality. The tasks that can be realized by the system can be also fulfilled, however usually in limited scope, by the class of commercial SCADA, DCS or similar systems that are available on the market. However, due to specialized system structure, there is a possibility to implement, apply and test modern and innovative techniques in the field of monitoring, advanced supervisory control, modeling and diagnostics of industrial processes which application in classical control and monitoring systems is difficult or even impossible.

Wide further system development is possible due to implemented plug-ins technology, open communication mechanisms and the possibility to introduce user-defined types of variables and their values (on each stage of system operation). This development can be realized not only by the research centers that took part in system creation but also by other, independent units.

The system structure was designed taking into account wide range of its possible fields of application. The use of the software platform and elaborated packages is planned not only in commercial applications but also in research-development and didactic tasks. The development of new system packages is conducted all the time. In respect to commercial application it is planned to used the system to realized the set of simulators of power generation units of conventional power stations. It is planned to use that simulators in the process of training the operators and other power station technical stuff. In the field of research and development the application of a system as monitoring system of a gas network is currently realized. The didactic use of the system is realized all the time. The system is used during the laboratories and project of courses related with the problems of process modeling (physical as well as based on process data and parametric models), diagnostics, automatic-control and applications of artificial intelligence methods in automatic-control.

ACKNOWLEDGMENTS

This work was supported in part by the Polish National Committee of Research under Grant R01 012 02 (DIASTER).

AUTHORS

Michał Syfert, Paweł Wnuk*, Jan Maciej Kościelny - Politechnika Warszawska, Instytut Automatyki i Robotyki, ul. Św. A. Boboli 8, Warsaw, 02-525, Poland.
E-mails: m.syfert@mchtr.pw.edu.pl,
p.wnuk@mchtr.pw.edu.pl,
jmk@mchtr.pw.edu.pl.
* Corresponding author

References

[1] M. A. Brdyś, P. Tatjewski, *Iterative Algorithms for Multilayer Optimizing Control,* Imperial College Press, London, 2005.

[2] J. Chen, R. Patton, *Robust model based fault diagnosis for dynamic systems,* Kluwer Akademic Publishers, Boston, 1999.

[3] W. Cholewa, J. Kiciński, Technical diagnostics. Inverse diagnostic models. Wydawnictwo Politechniki Śląskiej. Gliwice, 1997. (in Polish)

[4] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, Inc. New York-Basel-Hong Kong, 1998.

[5] K. Janiszowski, Identification of parametric models in examplesc. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2002. (in Polish)

[6] J. Korbicz, J. M. Kościelny, Z. Kowalczuk, W. Cholewa (ed). *Fault Diagnosis. Models, Artificial Intelligence, Application,* Springer, 2004.

[7] M. Kościelny, M. Syfert, P. Wnuk, *Advanced monitoring and diagnostic system "AMandD"*. Problemy Eksploatacji, vol. 61, no. 2, 2006, pp. 169-179 .

[8] J. Kościelny, M. Syfert, P. Wnuk, Advanced monitoring and diagnostic system "AMandD". In: *Safeprocess2006 A Proceedings Volume from the 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, P.R. China, August 29-September 1, Elsevier, vol. 1, 2006, pp. 635-640.

[9] W. Moczulski, *Technical diagnostics. The methods of knowledge acquiring*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2002. (in Polish)

[10] R. Patton, P. Frank, R. Clark (Eds.), *Issues of fault diagnosis for dynamic systems*. Springer, 2000.

[11] M. Syfert, P. Rzepiejewski, P. Wnuk, J. M. Kościelny, Current diagnostics of the evaporation station. In: *16th IFAC World Congress*, Praga, 4th-8th July, 2005.

[12] P. Tatjewski, *Advanced Control of Industrial Process. Structures and Algorithms*. Springer, 2007.