Advanced Communication Diagnostics in ISES Remote Experiment

M. Gerža, F. Schauer, K. Vlček

Abstract **-** This paper focuses on the analysis and diagnosis of communication between clients and the physical hardware of remote laboratories based on ISES (Internet School Experimental System). Many different types of connections are allocated to use comfortably the real-time experimenting in a remote laboratory. Each ISES unit includes some communication mechanism to provide data to underling or superior unit to process it in a given way. The main reason for proper communication is to deliver control commands from clients to the physical hardware to perform preset tasks in order to observe, measure and obtain real physical or electrical phenomena in form of data. The ISES remote experiment provides clients a unique educational tool for the purpose of the desired phenomena understanding. This tool is particularly useful for distant students, who are often hampered to attend experimental courses.

In the first chapter, the state of the art of the ISES remote experiment concept is introduced. In the next chapter, the basic analysis of the data communication, necessary for a cooperation of the units involved, i.e. physical hardware, Measureserver®, Internet and client, is presented. Further chapter proposes improvements of the most decisive communication segments. Three advanced diagnostic systems are introduced and used to prevent or significantly reduce occasional faults and anomalies. The first is the internal unit diagnosis, solving various communication faults inside the ISES remote experiment. The second one acts as the network traffic diagnosis, dealing with the detection, identification and quantification of anomalies, which can create congestion in network and may have an ill effect on the administrators or clients. As the third one, the cognitive fault diagnosis system is described with the aim of monitoring the distributed sensor network. It makes advantage of spatial and temporal relationship among sensor units connected to the ISES physical hardware to give sufficient information for failures reduction or avoidance. The last chapter summarizes all benefits of these diagnostic systems for ISES remote experiments reliability.

Keywords **-** ISES, Measureserver®, physical hardware, remote experiment, communication protocol, transmission, diagnosis

I. Introduction

THE traditional methods of teaching, oriented on students at secondary schools and universities, are quite obsolete and not so broadly popular to understand taught scientific themes. The contemporary students demand higher level teaching methods, which help them to perceive phenomena in better way in the field of physics, biology, chemistry and electro-engineering. Educational materials accessibility is important as well, especially for distant students who often prefer studying scientific themes via the Internet on their computers. These coveted advantages are provided by a remote experiment (RE) called e-laboratory. RE is built on ISES, which has been developed for educational purposes. The ISES is a complex tool for real-time operation, data acquisition, data processing and controlling physical hardware (HW). It is an open system consisting of the basic ISES hardware and ISES WIN software intended for a local experiment but it also has an option for the remote connection called ISES WEB Control Kit available anytime and anywhere.

The RE based on ISES WEB Control Kit is perceived like the superstructure so called ISES remote experiment (ISES RE), which has been developed by Charles University in Prague. After some time, the ISES RE has been significantly improved on a higher level educational tool by Tomas Bata University in Zlín in cooperation with Charles University in Prague so called EASY REMOTE - ISES (ER-ISES) in order to simplify settings and usage for teachers.

The ISES REs are categorized to several groups according to their complexity and the level of control as the basic, complex and scientific. Each RE consists of five cooperative units like is the physical HW (apparatus consisting of the ISES panel, meters, sensors and specific experimental devices generating given phenomena), Measureserver®, ImageServer, WebServer and WebClient. More technical details about the ISES RE are available in [1][2][15][16] and [17]. The clarifying scheme, including the communication relationships, is presented in Fig. 1.

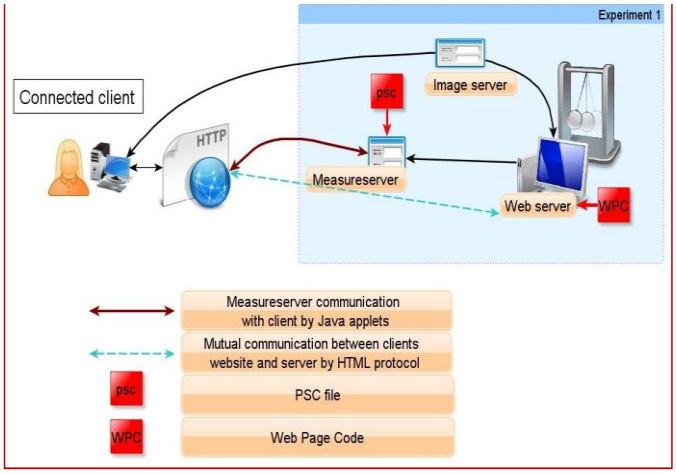

Fig. 1 Arrangement of the ISES remote experiment [3]

II. State of the Art

The ISES units dispose of adequate communication mechanisms to cooperate with neighboring units to deliver

requested information. Since many different types of data (signals and packets) are being processed and transmitted, so the functional concept implements the signal converting process and communication protocols.

A. *Physical hardware*

A low-level communication based principle is used between the ISES RE physical hardware modules and the AD/DA (Analog-to-Digital / Digital-to-Analog) convertor - 12 bits, time of conversion - 0.01 ms, installed as the PCI 1202 interface card inside an administrative computer. This device converts a continuous physical quantity (voltage) to a digital number that represents the quantity's amplitude and performs the inverse operation back to the physical quantity. All the used modules of physical HW, including the specific sensors and AD/DA convertor, are demonstrated in Fig. 2.

Fig. 2 ISES remote experiment including the AD/DA convertor card and a broad range of involved meters, sensors and probes [4]

B. *Measureserver® unit*

The Measureserver® (MS) is a significant software part of the ISES RE concept. It is perceived as a communication mediator between the physical HW and remote clients. The MS is constructed as the mathematical model used for designing of control programs by an external PSC program file to build a control and measurement logic.

Towards the physical HW, the MS communicates in reality with a software driver of the AD/DA convertor. It is entirely digital process based on reading data (values) directly from particular pins and writing data to respective pins which are translated by the AD/DA convertor. These pins are perceived as the inputs and outputs located on the ISES board that allows connecting particular measuring sensors and devices into the system. The low-level operation is always ensured by the PCI1202CardPlugin.ldp plug-in (intercommunication driver) loaded by the ScriptablePlugin2.ldp plug-in, which exploits the first one for its internal functioning. The ScriptablePlugin2.ldp plug-in builds the ISES RE logic by a PSC program delivered to the system by a responsible administrator. This plug-in is able to load any intercommunication file within the MS startup, but presently, the PCI1202CardPlugin.ldp is only available. The CFG configuration file, as a necessary part of MS intended for initial settings includes a reference to the ScriptablePlugin2.ldp plug-in. A scheme of the data communication relationships among particular modules is described in Fig. 3.

When data (control and measurement commands) come from a remote client to MS, the communication is realized by the TCP/IP (Transmission Control Protocol / Internet Protocol) protocol via the Internet and then goes to Intranet (local area network) in a building where the laboratory with ISES RE resides. Such the communication strictly requires a static public IP address of a computer hosting the MS and other important supporting services.

All commands, incoming from the entered client and physical HW on the other side, are processed by the deterministic way in a finite-state machine (FSM) realized by two involved parser mechanisms.

The first is the LR(1) parser that processes commands from the CFG configuration file for the purpose of the GUI (graphical user interface) setting. This parser is based on the static state transition tables called parsing tables, which codify the language grammar. These parsing tables are parameterized together with a lookahead terminal (lookahead establishes the maximum incoming tokens that the parser can use to decide, which rule it should use). More technical details, including several parsing examples, are available in [5] and [6].

The second one is the Recursive descent parser processing commands from the PSC program file in order to create internal data structures for ISES RE. The parser uses a general form of top-down parsing where backtracking may be involved. The parsing principle is based on the walking through a tree. More details, with a parsing example, are available in [7] and [8].

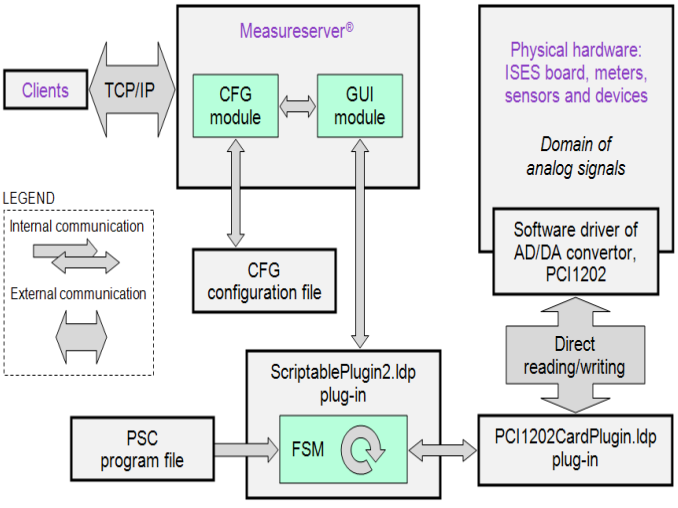

Fig. 3 Communication relationships among particular modules representing the ISES remote experiment

C. *Web server*

This unit is called Nginx that comes into the process when client enters a web page of the ISES RE by typing an IP

address or URL (Uniform Resource Locator) in any web browser (e.g. Firefox Mozilla, MS Explorer). Nginx is an open source reverse proxy server for HTTP, HTTPS, SMTP, POP3 and IMAP protocols and a web server [9].

When a client enters the ISES RE, the Nginx starts negotiating with the client and establishes a direct data communication between the MS unit and Java applet called ConnectionHub. Every applet, imported on the web page by the Nginx, uses services of the ConnectionHub to communicate with the physical HW via the MS unit.

D. Data network

As mentioned previously, the ISES RE uses communication protocols to negotiate with clients. When a client enters an URL of the ISES RE in a web browser to reach the physical HW, the communication starts by using the Internet Protocol Suite. After establishing the connection, initial packets enter a local area network (LAN) in the building where the laboratory resides. In the LAN, the connection is realized by Ethernet to communicate with the MS unit.

E. Client's interface

It is only one interface the clients can access the ISES physical hardware, therefore the web page's design and serviceability play important role as presented in Fig. 4.

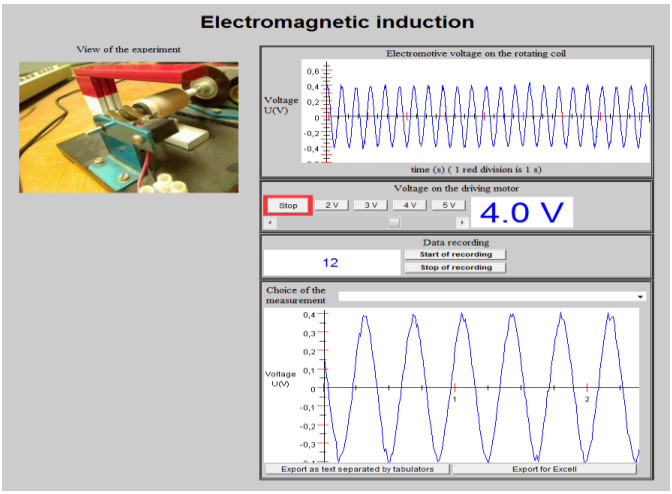

Fig. 4 Web page of the ISES RE presenting the Faraday's law of electromagnetic induction [4]

III. ISES REMOTE EXPERIMENT DIAGNOSIS

A diagnostic system poses an important part of every modern software and hardware application. Contemporary applications became too complex and they communicate usually with different subsystems, therefore administrators and clients should have a comfortable diagnostic tool to maintain functioning of such the applications.

A. Internal units diagnosis

The ISES RE concept has many deficiencies related to the communication among particular units. The most problematic point seems to be between the MS and physical HW where the involved AD/DA convertor is important.

Presently, the MS sometimes loses connection with the ISES RE or even stops its functioning. This is a serious problem that always has to be solved by the intervention of an administrator by experience-based actions (e.g. restart of the MS, re-connection of individual hardware modules). A solution is to deploy an intelligent diagnostic system intended for the communication that should primarily eliminate all the administrator's actions because a human factor can negatively influence the ISES RE functioning. The diagnostic system will be automatically monitoring and evaluating the internal connection between the MS and the AD/DA convertor. In case of the miscommunication, an alarm report will be generated and dispatched to the remote laboratory management system (RLMS), as a new unit of the improved ISES RE. The RLMS will be acting as a supervisor authorized to restart the MS as well to recover its initial functioning. Furthermore, the preset communication between the PCI1202CardPlugin.ldp plug-in and modules (e.g. ampere-meter, voltmeter), installed on the ISES board, will dispose of the robust self-checking mechanisms. When e.g. a cold link occurs in the connector linking the pin with module, a generated alarm report will be delivered to the RLMS to inform an administrator and entered clients about existing problem that obstructs the experimenting. The scheme, shown in Fig. 5, presents a deployment of the internal units diagnosis into the MS communicating with the ISES RE.

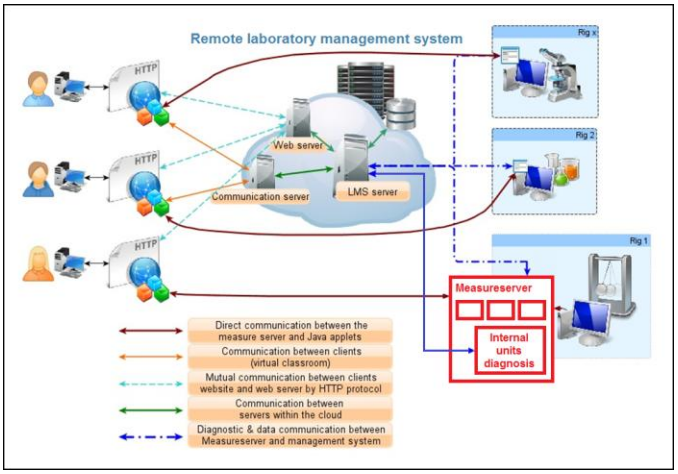

Fig. 5 Arrangement of the improved ISES RE based on the remote laboratory management system and the internal units diagnosis [3]

B. Network traffic diagnosis

As the second problematic point appears network traffic anomalies decelerating or blocking the communication between clients and the ISES RE. Anomalies are unusual and significant changes in network's traffic levels, which can often span multiple links. It is an important problem to understand the nature of traffic anomalies in a network. Regardless of whether the anomalies are malicious or unintentional, it is needed to analyze them for the following reasons:

- Anomalies can create congestion in the network and stress resource utilization in a router, which makes them crucial to detect from an operational standpoint.

- Some anomalies may not necessarily impact the network but they can have a dramatic impact on responsible network administrators or end clients.

It is a really difficult problem to solve because anomalous patterns must be extracted and interpreted from large amounts of high-dimensional noisy data. Hence, a general method is used to diagnose such anomalies. This method is based on a separation of the high-dimensional space occupied by a set of network traffic measurements into disjoint subspaces corresponding to normal and anomalous network conditions. The separation can be effectively performed by the coordinate transformation method called Principal Component Analysis.

An analysis of volume anomalies can be realized by using simple traffic measurements only from data links. The involved diagnostic method is able to:

1) detect when a volume anomaly is occurring,
2) identify the underlying origin-destination flow, which is the source of the anomaly,
3) estimate the amount of traffic involved in the anomalous origin-destination flow.

The introduced method is able to diagnose both existing and synthetically injected volume anomalies in real traffic in two networks. It diagnoses the largest anomalies and does so with a very low false alarm rate [11].

1) Volume anomalies

A typical network (e.g. backbone) is composed of nodes (also called Points of Presence or PoPs) that are connected by links. An Origin-Destination (OD) flow is define as the traffic that enters the network at the origin PoP and exits at the destination PoP. The path followed by each OD flow is determined by the routing tables. Therefore, the traffic observed on each network link arises from the superposition (two signals are added together) of these OD flows.

The volume anomaly term refers to a sudden (with respect to time step used) positive or negative change in an OD flow's traffic. Since such an anomaly originates outside the network, it will propagate from the origin PoP to the destination PoP.

A technique is used for diagnosing the volume anomalies. If a volume anomaly propagates through the network, it should be observed on all links it traverses. Anomalies based on the OD flow are identified by observing only link counts.

The diagnosis difficulty stems in part from the fact that it uses only link data, which can be collected via SNMP (Simple Network Management Protocol). Necessary inferences must be formed about unusual events occurring in the underlying OD flows from these link data.

Examples of this difficulty are presented in Fig. 6. The top plot on each side of the figure shows an OD flow time series with an associated volume anomaly - this information is not available to the algorithms, but just to show the nature of these anomalies. The point at which each anomaly occurs is designated by a circle on the timeline. Below the timeline are plots of link traffic on the four links that carry the given OD flow. These four plots represent the data that is available to the algorithm. The diagnostic system processes link data to:

1) detect that at the time shown, the network is experiencing an occurred anomaly,
2) isolate the four links shown as those experiencing the detected anomaly,
3) estimate the size of the spike in the OD flow.

Three observations could be performed from these examples. First, while the OD flows have pronounced spikes, the corresponding spike in the link traffic is dwarfed, and difficult to detect even from visual inspection. For instance, the traffic volume at the spike time on links, defined as *c-d* and *b-c* in Example 1, is hardly distinguishable. Second, the temporal traffic patterns may vary substantially from one link to another. In Example 2, the *i-f* link has a smooth trend, whereas the other links for the OD flow have more noisy traffic. Separating the present spike from the noise in the traffic on the *c-b* link is visually more difficult than separating the spike in the *i-f* link. Thus isolating all the links exhibiting an anomaly is challenging. Finally, mean traffic levels vary considerably. In Example 1, the mean traffic level on the *c-d* link is more than twice that of the *f-i* link. The varying traffic levels makes it difficult to estimate the size of the volume anomaly and hence its operational importance.

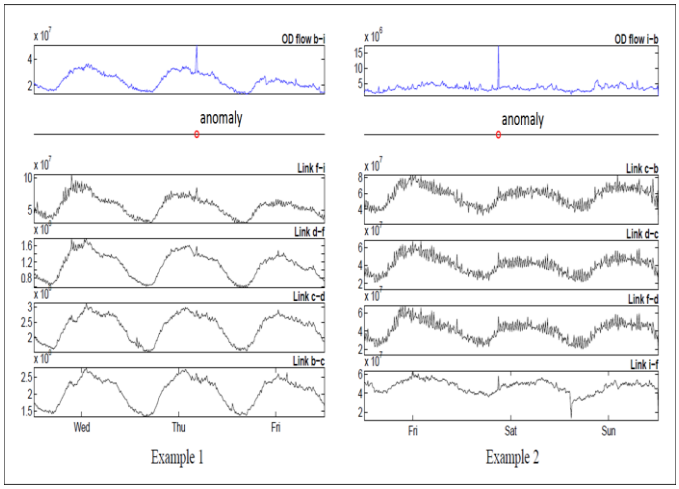

Fig. 6 Examples of present anomalies at the Origin-Destination flow level (top row) that is required to diagnose from link traffic [11]

The problem of diagnosing a volume anomaly in an involved OD flow can be separated into the following steps:

- *Detection* consists of designating those points in time at which the network is experiencing an anomaly. An effective algorithm for solving the detection has a high detection probability and a low false alarm probability.
- *Identification* consists of selecting the true anomaly type from a set of possible candidate anomalies. The method is extensible to a wide variety of anomalies. However, a first step, the candidate anomaly set is the set of all OD flows.
- *Quantification* is the problem of estimating the number of additional or missing bytes in the underlying traffic flows. The quantification is important because it gives a measure of the importance of the existing anomaly.

The diagnosis also requires the detection of an anomaly time, the identification of the underlying responsible OD flow and the quantification of an anomaly.

2) Data acquirement

The method operates on link traffic data obtained by SNMP. Traffic anomalies can last anywhere from milliseconds to hours. It can be used on data with any time granularity, e.g. to work with data binned on 10 minute intervals. Binning is a way to group a number of more or less continuous values to a smaller number of bins [10].

In order to validate data against true OD flows, a set of link traffic counts must be obtained consistent with sampled OD flow data collected from the network. To perform this, the traffic matrix estimation method is followed and a construction of the link counts is then performed from OD flow counts, which use a routing table taken from the network in operation.

3) Subspace analysis of link traffic

The diagnosis of anomalies in traffic requires the ability to separate them from normal network-wide traffic. In this subchapter, the Principal Component Analysis (PCA) is described to separate normal and anomalous network-wide traffic conditions.

The PCA is a coordinate transformation method that maps a given set of data points onto new axes. The axes are called the principal axes or principal components. When working with zero-mean data, each principal component has the property that it points in the direction of maximum variance remaining in the data, given the variance already accounted for in the preceding components. As such, the first principal component captures the variance of the data to the greatest degree possible on a single axis. The next principal components then each capture the maximum variance among the remaining orthogonal directions. Thus, the principal axes are ordered by the amount of data variance that they capture.

An illustration of the difference between normal and anomalous traffic variation is shown in Fig. 7, as captured in the PCA decomposition. The figure shows sample projections of the network 1 dataset onto selected principal components. On the left, projections onto the first two principal components (*u1* and *u2*) are presented, which capture the most significant variation in the data. These time series are periodic and reasonably deterministic and clearly capture the typical diurnal patterns, which are common across traffic on all links. Note that *u1* and *u2* are roughly 180 degrees out of phase, meaning that the two can be used in linear combination to roughly construct of sinusoid of any phase. Thus the extraction of common temporal patterns via the PCA does not require the underlying traffic time series to have the same periodic phase as reflected e.g. in traffic in the same time zone. The subspace method assigns the traffic variations to the normal subspace.

Presented Fig. 7 also shows projections *u6* and *u8*. In the contrast to involved *u1* and *u2*, these projections of the data exhibit significant anomalous behavior. These traffic spikes indicate unusual network conditions possibly induced by a volume anomaly at the OD flow level. The subspace method treats such projections of the data as belonging to the anomalous subspace.

A variety of procedures can be applied to separate the two types of projections into normal and anomalous sets. Based on examining the differences between typical and atypical projections, a simple threshold-based separation method has been developed to work well in practice. Specifically, a separation procedure examines the projection on each principal axis in order; as soon as a projection is found that exceeds the threshold, e.g. contains a deviation from the mean, that principal axis and all subsequent axes are assigned to the anomalous subspace. All previous principal axes then are assigned to the normal subspace. All the dimensions showing significant variance are assigned to the normal subspace when this procedure results in placing the first four principal components in the normal subspace in each case.

The traffic is decomposed on each link into normal and anomalous components after separating to the space of possible traffic measurements into the subspaces.

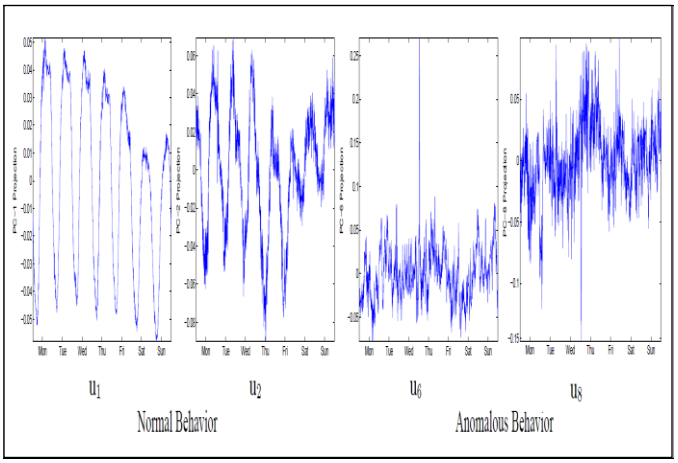

Fig. 7 Example of the projections onto principal components showing normal and anomalous traffic variation in network [11]

4) Diagnosing volume anomalies

The methods used for detecting and identifying volume anomalies draw from a theory developed for the subspace-based fault detection in multivariate process control.

Detecting volume anomalies in link traffic relies on the separation of link traffic at any time step into normal and anomalous components. They can be also called as the modeled and residual parts of the link traffic in a network.

The key idea in the subspace-based detection step is that, once both the normal and anomalous subspaces have been constructed, this separation can be effectively performed by forming the projection of link traffic onto these two subspaces.

As the next diagnostic step is a process of the identification. In the subspace framework, a volume anomaly represents a displacement of the state vector away from the normal subspace. This state vector is expressed as a sum of the sample vector for normal traffic conditions and the magnitude of the anomaly, which is influenced by the vector defining the manner in which this anomaly adds traffic to each link in the

network. The particular direction of the displacement gives information about the nature of the anomaly. Thus the approach to anomaly identification is to ask which anomaly out of a set of potential anomalies is best able to describe the deviation of state vector from the normal subspace.

When an estimation of the particular volume anomaly was formed, the last step comes into the process called quantification. This method is able to estimate the number of bytes constituting this anomaly.

5) Validating results

The validation is centered on answering two questions:

1) How well can the method diagnose actual anomalies observed in real data?
2) How does the time and location of the anomaly affect performance of the method?

The first question can be answered as follows. It is reached up by using the time series analysis on OD flow data to isolate first a set of true anomalies. This approach allows evaluating the subspace method quantitatively. In particular, it allows the measurement of detection and false alarm probabilities.

The second question can be answered as well. It is realized by injecting anomalies of different sizes in OD flows and applying a procedure to diagnose these known anomalies from link data. This is performed repeatedly for each time step and for each anomaly to form the picture of how diagnosis effectiveness varies with the time and location of the occurred anomaly in a data network.

In each case, the performance of each step must be quantified in the following diagnosis procedure. A detection success is measured by two metrics: the detection rate and the false alarm rate. The detection rate is the fraction of true detected anomalies. The false alarm rate is the fraction of normal measurements that trigger an erroneous detection. An identification success is captured in the identification rate, which is the fraction of detected anomalies that are correctly identified. Finally, a quantification success is measured by computing the mean absolute and relative error between the estimate and the true size of identified volume anomalies [11].

C. Cognitive fault diagnosis system

Let us find a solution for the operative mechanism of ISES Measureserver® with respect to its diagnostics to reduce faults coming from sensors of the RE physical hardware, using the artificial intelligence approach.

Sensors monitoring a real environment are prone to faults or aging, ill affecting ISES RE functioning, or even fallout of the whole apparatus. In turn, the permanent or transient faults can influence sensors functioning, causing errors in the RE processing chain. Such erroneous information may exert strong side effects on the subsequent control chain leading to bad decisions and inappropriate control actions.

Fault diagnosis system (FDS) should play an important role of supervising the process operations for the purpose of detecting, isolating and identifying a potential fault and design possible accommodation actions. The main components of FDS are derived from the comparison of the running and model data of functioning. Model data, often unavailable, is frequently substituted by the experimental data generated by the ISES physical hardware during or after the measurement.

When a change with respect to the model is detected by applied FDS, the following situations might arise [12]:

- Model change: The model is no longer representing the current data due to the model approximation deficiencies.
- Change in the environment: The environment is a time variable quantity and the trained model is no more able to explain the acquired data.
- Fault: The sensor or its electronic unit is affected by a fault inducing an error.

Existing FDSs intended for sensor networks do not generally allow distinguishing between occurred faults and environment changes. Moreover, in the original model bias is considered negligible, which is hardly acceptable hypothesis in many applications. However, a cognitive FDS may influence sensor data streams. This type of FDS already recognizes the model bias existence during measurements and proposes a method for discriminating between faults and changes.

Let us further attempt to propose a solution for FDS design, based on the artificial intelligence approach, introducing dependency graphs and information related to spatial and temporal relationships among sensor data streams.

1) Functional concept

Hidden a-priori information concerning spatial and temporal relationships among proposed sensor data streams is exploited, leading to a functional dependency graph where nodes are the used sensors and arcs are associated with the sensor-to-sensor functional relationship. In particular, for each sensor couple, Hidden Markov model (HMM) is designed which gives the parameters of linear time invariant (LTI) model approximating the relationship. As such, spatial redundancy is modeled with HMM in the parameter space of linear time invariant dynamic models, embedding the time dependency. When the likelihood between the HMM-based learning machine and the new incoming data stream falls below a preset threshold (which can be inserted by the teaching process), a change is detected by the HMM-based change detection test (CDT) at the detection layer. The cognitive layer of FDS, activated in response to a change, starts alarm raised by the CDT, discriminates time variant and bias faults using the dependency graph of the network. At the same time, it allows for isolating the fault for a possible accommodation phase [12].

In following we describe the theoretical solution of a cognitive FDS in order to diagnose all sensors installed in RE physical hardware to reduce possible faults coming from the measurement subsystems. The proposed FDS is intended to be built in the next generation of the Measureserver® unit for the AI on line diagnosis of the RE physical hardware.

2) Modeling functional relationships in sensor networks

Let us consider a sensor network composed of N fixed

sensing units, which are deployed within the environment *P*. Each unit can host up to *M* sensors giving information on various physical properties of *P* space (for example, temperature, humidity, vibrations, rain intensity). Each *j*-th sensor of the *i*-th unit acquires a scalar data stream $X_{i,j}$.

The FDS runs as a part of the MS unit situated in the control room where the RE physical hardware is installed [12].

a) Modeling the network: the dependency graph

The cognitive framework for the fault diagnosis relies on the ability to model functional relationships among the acquired information on the space *P*. In detail, each functional relationship captures spatial and temporal dependencies from data provided by a generic couple of involved sensors. Fig. 8 shows an example of the sensor network with dependencies.

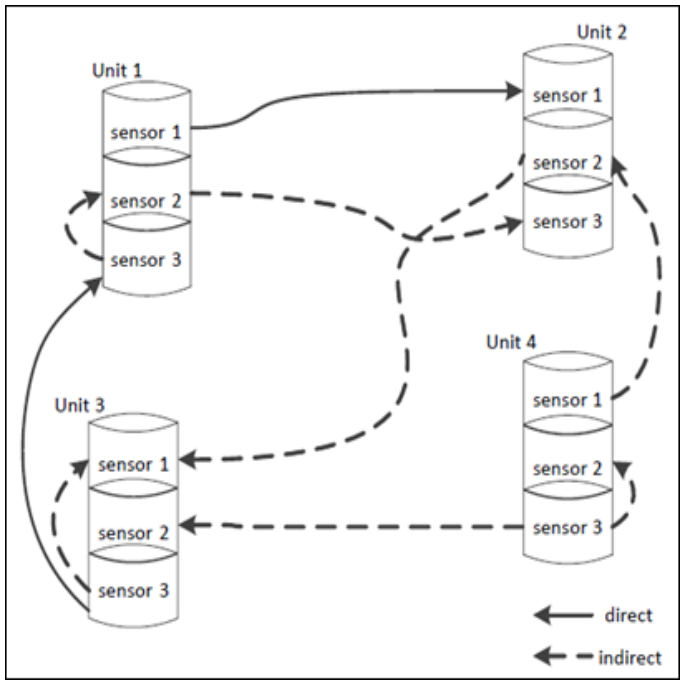

Fig. 8 Direct and indirect relationships in the network [12]

A direct relationship exists among couple of sensors of the same type like is usually temperature vs. temperature. If data streams $X_{i,j}$ and $X_{v,j}$, $i \neq v$ are correlated, then an arc linking the *j*-th sensor of unit *i* with its counterpart of unit *v* is introduced. For example, two clinometers insisting on the same connected structure are related; those deployed far apart probably are not. An indirect relationship can be introduced between two generic sensors by means of a third entity. Indirect relations are mitigated by the presence of compensation mechanisms. Information useful for the analysis must be extracted before compensation takes place.

In reality, direct and indirect relationships introduce a functional constraint among couples of sensors. Denote by *f{(i,j),(u,v)}* the functional relationship between the generic *j*-th sensor of unit *i* and the *v*-th sensor of unit *u*. The nodes of *G* are the network sensors where the arcs represent the relationships among couples of sensors. Given a network, not all the *(N×M)(N×M−1)* relationships in *G* are relevant.

The *reduced dependency graph* is then derived from *G* and defined as graph *GR={V,E}* where *V* is the set of nodes of the graph representing the *N×M* sensors and *E* a set collecting all arcs associated with functional relationships whose correlation is above a threshold. The level of dependency associated with relationship *f{(i,j),(u,v)}* is here chosen to be the linear correlation index between two data streams $X_{i,j}$ and $X_{v,u}$. We remove from G_R all the isolated nodes. Fig. 9 shows us the graph-based representation of the sensor network proposed in Fig. 8. We have 4 units; each unit is a sub-graph representing the sensors with bindings [12].

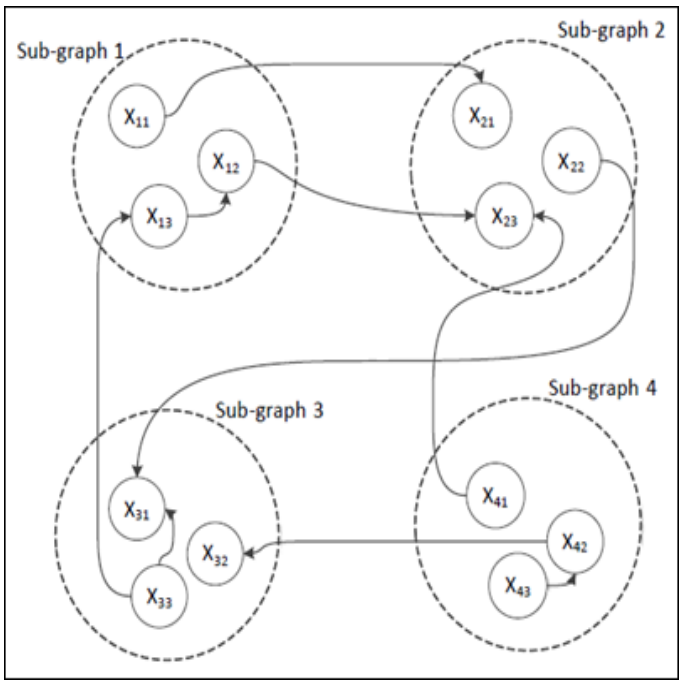

Fig. 9 The dependency graph of sensor network of Fig. 8 [12]

b) Modeling the functional relationship between two same type sensors by using Hidden Markov model

We assume that the relationship among couple of sensors *f{(i,j)(u,v)}* can be modeled either as a time invariant (TI) dynamic system or as a finite sequence of TI dynamic systems satisfying the HMM hypotheses.

Let us imagine to model a *f{(i,j)(u,v)}* with the Single-Input Single-Output (SISO) linear model. A given SISO locally approximates the output.

A training dataset is composed of N_T *{input, output}* couples and a loss function whose minimization provides us an estimation of the optimal parameter.

Under the assumption that each involved *f{(i,j),(u,v)}* function satisfies the exponential stability for closed loop.

It comes out that under the above assumption and a sufficiently large *N* the distribution underlying the parameter vectors is defined as the multivariate Gaussian, with a mean and covariance matrix *P*.

HMM with parameters ruled by a mixture of Gaussians becomes a natural solution to approximate *f{(i,j),(u,v)}*. The HMM nodes of the represent in reality a probabilistic ensemble of used LTI models minimizing the model bias if a training set is sufficiently informative. By used modeling

parameters with HMM, we mitigate the effect of model bias and time variance provided that the defined training set is sufficiently informative as well and it explores time variance and nonlinearity [12].

3) Cognitive fault diagnosis system

The FDS is organized as the two-layer architecture, shown in Fig. 10. The lower level is composed by a set of change detection tests (CDTs) observing the stationarity of a relationship associated with a couple of sensors in GR. Each HMM-CDT works in the parameter space to detect variations in the relationship between two involved sensors. The CDT is not able to distinguish among changes induced by a fault in a sensor, an environmental change in P or a false positive generated by a model bias since such classes are indistinguishable. To address this issue the upper level of the FDS has been designed to be able to discriminate among faults, changes in P and false positives by exploiting information associated with the network graph GR. The upper level of the FDS relies on the cognitive algorithm aggregating decisions and log-likelihood information provided by the HMM-CDT in the lower level [12].

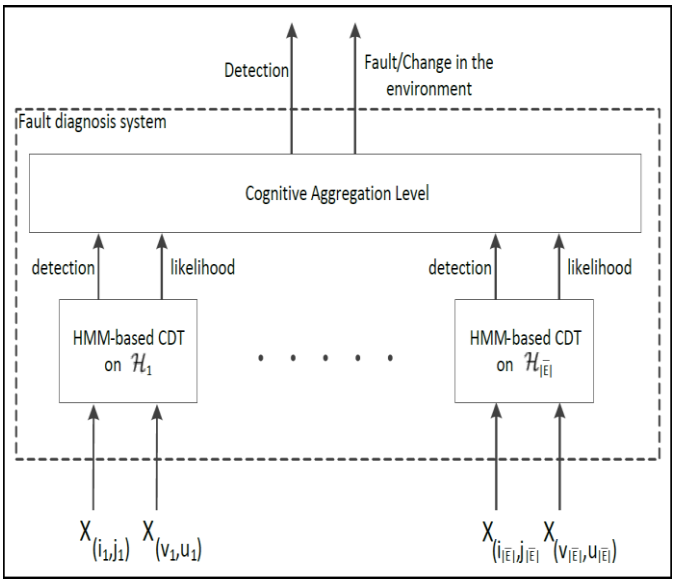

Fig. 10 Configuration of the proposed fault diagnosis system [12]

a) The HMM-based change detection test

The proposed HMM-CDT aims at evaluating, by means of HMM, the evolution over time of the estimated parameters approximating the relationship $f\{(i,j),(u,v)\}$; $X_{(i,j)}$ is the output and $X_{(u,v)}$ the input of the LTI. Estimated parameters are estimated on overlapping windows of N_T data.

The HMM-CDT requires training of HMM devoted to model the relationship between sensors *(i,j)* and *(u,v)* is trained by the Baum-Welch algorithm [13].

During the operational life, the parameter is estimated on the *s*-th window of data and the log-likelihood that is computed with the Viterbi algorithm [14].

When the log-likelihood decreases below a threshold T_h, a change in the relationship is detected (the sequence of inputs is no more recognized by the learning machine). The threshold T_h can be defined by a responsible operator who is able to exploit a-priori available information [12].

b) The cognitive aggregation level

The cognitive level aggregates the information coming from all sensor units to distinguish among faults, changes in *P* and false positives induced by model bias in the HMM-CDT. Differently from the HMM-CDTs executed sequentially, the cognitive aggregation level is activated only in response to a detection alarm raised by at least HMM-CDT. Detections and log-likelihoods of others CDTs are used to assess and, possibly, identify the change.

The motivating idea is that a change in *P* for a given type of sensors must be also perceived by a set of other CDTs, at least as a decrement in the log-likelihood values, which are not necessarily below the threshold. Differently, in the case of faults, only the CDTs associated with relationships that have either as input or output the faulty sensor are affected by the change. Finally, if a false positive occurs, other involved CDTs should not be affected.

To evaluate the reliability of the information coming from HMM-CDTs we introduce a reliability index for the HMM.

Weights are computed on the training set; the *weighted reduced graph* is the reduced graph that is augmented with the weight information.

Definitions are constructed as follows:

- Let E^+ be the set of functional relationships such that either the source or the target node of the arc is used $X_{(i,j)}$.
- Let E^- be the set of functional relationships such that either the source or the target node of the arc is used $X_{(v,u)}$.
- Let E^P be the set of functional relationships whose defined source or target node is neither $X(i_q,j_q)$ nor $X(v_q,u_q)$.

After a change detected in $f\{(i,j)(v,u)\}$ the remaining functional relationships of the weighted reduced dependency graph are partitioned into sets E^+, E^- and E^P. The reason for the partitioning is described as follows:

- a fault in sensor $X(i,j)$ affects the functional relationships in E^+ but not in E^- and E^P,
- a fault in sensor $X(v,u)$ affects the functional relationships in E^- but not in E^+ and E^P,
- a change in *P* affects the functional relationships in all groups, in E^-, E^+ and E^P,
- a model bias, affecting HMM, would mostly affect the relationship between *(i,j)* and *(u,v)* but not the relationships in E^-, E^+ and E^P provided that approximating relationships are characterized by different bias contributions.

An example of partitioning is shown in Fig. 11a; in Fig. 11b a change is detected in functional relationship $f\{(3,3)(1,3)\}$. The definitions are used as follows:

$$E^+ = \{f_{\{(3,3)(3,1)\}}\};$$
$$E^- = \{f_{\{(1,3)(1,2)\}}\};$$
$$E^P = \{f_{\{(1,1)(2,1)\}}, f_{\{(1,2)(2,3)\}}, f_{\{(3,1)(2,2)\}}, f_{\{(4,1)(2,3)\}}, f_{\{(3,2)(4,2)\}}, f_{\{(4,3)(4,2)\}}\}.$$

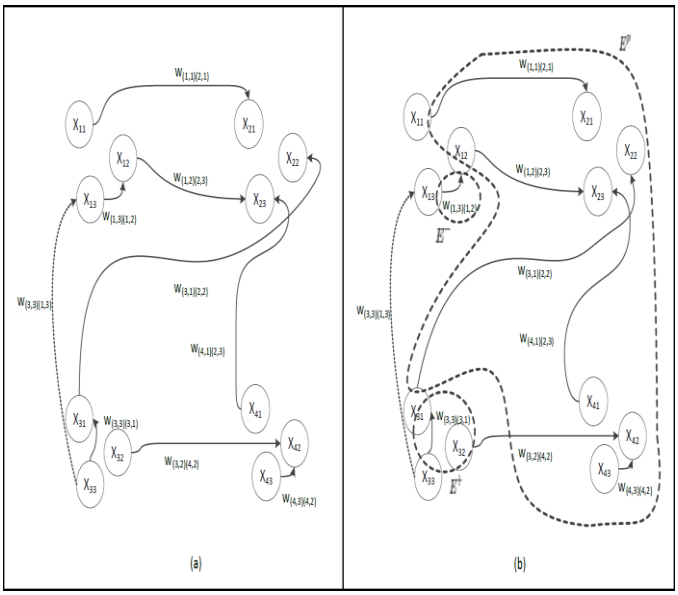

Fig. 11 The proposed cognitive aggregation level: a) the reduced weighted dependency graph; b) an example of arcs partitioning into defined groups E^+, E^- and E^P given a change detected in the functional relationship *f{(3,3),(1,3)}* [12]

Defined index of the data window where the HMM-CDT detected a change, the proposed aggregation level computes the normalized sum of the log-likelihoods, suitably weighted, of the arcs in E^+, E^- and E^P.

The core of the cognitive aggregation level is thus the ability to compute S^+, S^- and S^P by exploiting information coming from all the relationships of the weighted reduced dependency graph. S^+, S^- and S^P measure how the change detected in the functional relationship *f{(i,j)(v,u)}* is perceived in other relationships. If a fault affects sensor *(i,j)*, S^+ should decrease, while S^- and S^P should not. Similarly used, if a fault affects sensor *(u,v)*, S^- should decrease, S^+ and S^P not. If a change in P occurs, S^P should decrease as well as S^+ and S^-.

To detect decreases in S^+, S^- and S^P we rely on a simple thresholding mechanism that calculates thresholds T^+, T^- and T^P which can be scaled by a coefficient factor specified as c_2 to increase the robustness with regard to false positives. We suggest selecting a condition as $c_2 > c_1$ since we want to detect decreases in the likelihood that did not yet raised an alarm. In reality, if we consider $c_2 \leq c_1$, then we would require that the weighted average of the likelihoods decreases below the weighted average of the thresholds for change detection T_hs but this is nonsense since relationships in E^+, E^- and E^P did not detect a change yet.

To sum up, the cognitive aggregation level acts as follows:

- If S^P decreases below threshold T^P, a change in P is successfully identified.
- If $S^P > T^P$ and $S^+ < T^+$ (or $S^- < T^-$), a fault in sensor $X_{(i,j)}$ (or in $X_{(v,u)}$) is effectively detected.
- If $S^P > T^P$ and $S^+ > T^+$ and $S^- > T^-$, a false positive which is induced by a model bias is detected.

If both S^+ and S^- are above their respective thresholds, we can raise the alarm fault in either $X_{(i,j)}$ or $X_{(v,u)}$ but we cannot isolate the affected sensor since not enough information is available in the system [12].

IV. Functional Benefits

The introduced diagnostic systems provide us an efficient solution how to avoid or reduce possible faults coming from some sensors and modules in the ISES RE. The second benefit is an elimination of the congestion in a network caused by wide traffic anomalies when clients make experimenting.

These diagnostic systems should cooperate with the RLMS that performs accommodation actions based on scenarios in case of detected and identified ill events occurred during the experimentation. The scheme, presented in Fig. 12, shows an implementation of the diagnostic systems inside the MS unit.

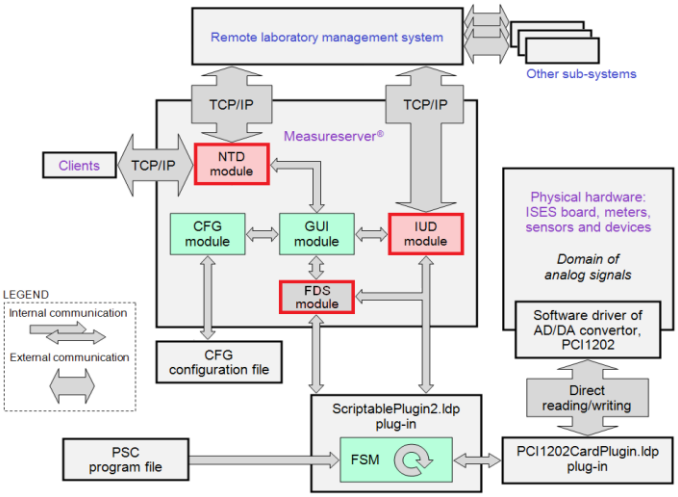

Fig. 12 Implementation of the internal units diagnosis (IUD) together with the network traffic diagnosis (NTD) and the fault diagnosis system (FDS) in the Measureserver® unit

V. Conclusions

This paper has presented the analysis of advanced communication among particular units of the ISES remote experiment, and provided you possible improvements by using three different diagnostic systems. It has been the objective of our work to analyze the low-level communication to understand its basic principles. The further part of this objective has been focused on diagnosing occasional faults occurred during the communication of separate ISES hardware modules and involved sensors. It includes the detection, identification and quantification as well intended for existing wide traffic anomalies in a network. We have analyzed the suitable diagnostic approaches to implement into the ISES remote experiment to avoid or significantly reduce such ill events at process time of experimenting.

Our conclusions may be formulated as follows.

1) The experimentation based on the ISES remote experiment is a new approach of teaching and learning in comparison with traditional forms of education.
2) The Measureserver® is a core unit of the ISES remote experiment responsible for communicating between clients and physical hardware modules.
3) The internal units diagnosis is a suitable approach for monitoring and evaluating the internal communication between the Measureserver® and the AD/DA convertor to

avoid faults coming from the ISES physical hardware.

4) The network traffic diagnosis fits to the wide traffic anomalies occurring in a network for the purpose of detecting, identifying and quantifying them, and to report ill events to responsible administrators and active clients using the ISES remote experiment.
5) The cognitive fault diagnosis system is an advanced approach integrated into the Measureserver® to avoid occasional faults coming from sensors as the components of the ISES physical hardware, which can negatively affect sometimes the ISES remote experiment functioning.
6) We also plan an improvement of these diagnostic systems concerning intelligent corrections performed when faults or anomalies come into the system.

ACKNOWLEDGMENT

The paper was published thanks to the Grant of the Internal Agency of Tomas Bata University No. IGA/FAI/2014/044, and partially by the Grant of the Kega Agency KEGA Agency projects No 011TTU-4/2012 and 020TTU-4/2013 and Grant of the agency APVV project No. APVV 0096-11.

REFERENCES

[1] ZEMAN, Petr. Software environment for integration of measured data from remote laboratory and simulation. Ostrava: VŠB-Technical University of Ostrava, 2012.
[2] ZEMAN, Petr. Software environment for control of remote experiments. Ostrava: VŠB-Technical University of Ostrava, 2011.
[3] KRBEČEK, Michal. ISES remote experiments configuration. Zlín. UTB ve Zlíně, Fakulta aplikované informatiky, 2013.
[4] LUSTIG, František. Internet School Experimental System iSES [online]. Prague, Czech Republic, 2009 [cit. 2014-06-05]. Available: http://www.ises.info/index.php/en
[5] Canonical LR parser: Constructing LR(1) parsing tables. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014 [cit. 2014-06-05]. Available: http://en.wikipedia.org/wiki/Canonical_LR_parser
[6] Parsing: Lookahead. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014 [cit. 2014-06-05]. Available: http://en.wikipedia.org/wiki/Parsing#Lookahead
[7] REDDY. Top-Down Parsing, Recursive-Descent Predicative Parsing [online]. [cit. 2014-06-05]. Available: http://www.facweb.iitkgp.ernet.in/~niloy/Compiler/notes/TDP.doc
[8] WILKINSON, Leland. Recursive Descent Parser. [online]. 2008 [cit. 2014-06-05]. Available: http://www.cs.uic.edu/~wilkinson/Applets/parser.html
[9] Nginx. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014 [cit. 2014-06-05]. Available: http://cs.wikipedia.org/wiki/Nginx
[10] Binning. Spotfire Technology Network [online]. TIBCO Spotfire, 2013 [cit. 2014-05-06]. Available: http://stn.spotfire.com/spotfire_client_help/bin/bin_what_is_binning.htm
[11] LAKHINA, Anukool, Mark CROVELLA a Christophe DIOT. Diagnosing network-wide traffic anomalies. In: SIGCOMM '04 Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications. New York, NY, USA: ACM, 2004, s. 12. ISBN 1-58113-862-8. DOI: 10.1145/1015467.1015492. Available: http://www.cs.cornell.edu/people/egs/cornellonly/syslunch/fall04/anomalies.pdf
[12] ALIPPI, Cesare, NTALAMPIRAS, Stavros a ROVERI Manuel. A Cognitive Fault Diagnosis System for Distributed Sensor Networks. In: Neural Networks and Learning Systems: IEEE Transactions on (Volume: 24, Issue: 8). Milan, Italy: Dipt. di Elettron. e Inf., Politec. di Milano, 2013, 1213 - 1226. ISSN 2162-237X. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6502725
[13] RABINER and JUANG, An introduction to Hidden Markov models, IEEE ASSP Magazine, pp. 4-15, January 1986.
[14] DURBIN, EDDY, KROGH and MITCHISON, Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, July 1998. Available: http://www.bioinfo.org.cn/~wangchao/maa/Durbin_et_al_Biological_Sequence_Analysis_CUP_2002_no_OCR.pdf
[15] KRBEČEK, Michal, František SCHAUER and Karel VLČEK. Communication Requirements of Laboratory Management System. In: LATEST TRENDS on SYSTEMS - VOLUME II: Proceedings of the 18th International Conference on Systems (part of CSCC 2014), Santorini, Greece, 2014, p. 686-691. ISBN 978-1-61804-244-6, ISSN 1790-5117. http://www.europment.org/library/2014/santorini/bypaper/SYSTEMS/SYSTEMS2-56.pdf
[16] Hamid, R., and Syakirah Afiza Mohammed. 2010. Remote Access Laboratory System for Material Technology Laboratory Work. In International Conference on Engineering Education and International Conference on Education and Educational Technologies - Proceedings, 311–16. http://www.scopus.com/inward/record.url?eid=2-s2.0-79958730131&partnerID=tZOtx3y1
[17] Drigas, A. S., J. Vrettaros, L. G. Koukianakis, and J. G. Glentzes. 2006. A Virtual Lab and E-Learning System for Renewable Energy Sources. WSEAS Transactions on Computers 5 (2): 337–41. http://www.scopus.com/inward/record.url?eid=2-s2.0-33645137921&partnerID=tZOtx3y1

M. Gerža, **F. Schauer** and **K. Vlček** are with the Tomas Bata University in Zlín, Faculty of Applied Informatics, Nad Stráněmi 4511, Zlín, 760 05, Czech Republic. **F. Schauer** is also associated with University of Trnava, Faculty of Education, Priemyselná 4, 917 01, Trnava, Slovak Republic (email contacts: michal.gerza@email.cz, fschauer@fai.utb.cz, vlcek@fai.utb.cz).