World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:5, 2007

A Method for Analysis of Industrial Distributed Embedded Systems

Dmitry A. Mikoyelov

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf

***Abstract*—**The paper presents a set of guidelines for analysis of industrial embedded distributed systems and introduces a mathematical model derived from these guidelines. In this study, the author examines a set of modern communication technologies that are or possibly can be used to build communication links between the subsystems of a distributed embedded system. An investigation of these guidelines results in a algorithm for analysis of specific use cases of target technologies. A goal of the paper acts as an important base for ongoing research on comparison of communication technologies. The author describes the principles of the model and presents results of the test calculations. Practical implementation of target technologies and empirical experiment data are based on a practical experience during the design and test of specific distributed systems in Latvian market.

***Keywords*—**Distributed embedded system, analytical model, communication technology.

I. Introduction

THE goal of the paper is to specify a set of rules necessary to design a mathematical analytical model intended for evaluation of communication technologies in distributed embedded systems and the model itself. The paper introduces an intermediate stage of the main research of the author within the PhD promotional study domain. A mathematical model for analysis of communication technology implementation in distributed embedded systems is introduced.

The paper presents a set of rules for further research and development of a methodology for analysis of communication schemes in distributed embedded systems. This material introduces notations for ongoing development of an analytical model, which is intended to identify the most appropriate technology for a given distributed system (a case study) implementing one of modern communication technologies referenced in the paper.

The first part of the paper consists of theses, which are thematically grouped into separate sequential chapters. The purpose of the style implemented in the paper is to provide a clear view on the problem domain and author's thoughts on the steps to develop the end solution. The second part of the paper is dedicated to the description of the introduced model and a demonstration of its implementation on a test use case. The paper is based on previous research [1-4], assisting in completion of the mathematical model.

The target solution in this case is a method for an automated selection of the most appropriate technology (or a set of close technologies) in a given environment (distributed embedded system description and additional specific customer requirements), implementing an analytical mathematical model with a target function. The target function is intended to be based on values of a number of unique factors. In turn, every factor is based on a set of coefficients specific to each factor individually. The values of coefficient variables are calculated by processing the measurement data basing on [5-10].

The Background of the Research

The main goal of industrial process automation is centralized monitoring and control center with independent control subsystems in each remote location. This increases the number of remote subsystems, which need communication middleware. The second goal of industrial process automation is reduction of industrial system maintenance costs. This includes getting the current status, updating firmware, and making changes to the action sequence the subsystem performs.

There are numerous more or less independent critical attributes of embedded telecommunication systems, which should be implemented in a distributed embedded system. Speaking of industrial process automation, these attributes immediately become much more critical comparing to home or office communication requirements. Downtime on a factory floor will affect in a way of enormous financial expenses. In some processes, which are related to substances with low viscosity (oil fractions, diesel or black oil, for example), the heating of the sub-product must be constant, as well as its transportation though the processing line. Otherwise, the valuable equipment will malfunction for a long time or become unusable at all until replacement. Distributed network sites operate at a downtime cost of $20000 to $80000 per hour, with companies like stock firms impacted at rates of $6 million per hour. Even at $80K per hour, an average downtime of 88,6 hours calculates to $7,1 million (Strategic Research Corporation.)

It is true that all these manufacturing process risk factors have to find answers in a rapid-action, reliable monitoring and control system, which consists of data transfer links and embedded components. Let us analyze the most important critical attributes that are applied to the distributed embedded system and its telecommunication subsystems in industrial process automation.

Manuscript received June 30, 2007. This work has been partly supported by the European Social Fund within the National Programme "Support for the carrying out doctoral study programm's and post-doctoral researches" project "Support for the development of doctoral studies at Riga Technical University".

D. A. Mikoyelov is with the Faculty of Computer Science and Information Technology, Institute of Applied Computer Systems, Department of Applied Computer Science, Riga Technical University, Meza 1/3, Riga LV 1048, Latvia (e-mail: dmitry.mikoyelov@gmail.lv).

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf

In fact, the need for high *integrity* in almost all distributed embedded systems is obvious, but how to ensure it is less obvious. Modern approaches to designing reliable systems require knowledge of all subsystems of a system – knowledge that cannot be ensured in the rapidly changing environments in which distributed embedded system will be integrated.

Evaluation mechanisms that apply to standard networks of data-processing devices may well fail to apply in the context of distributed embedded systems, where subsystems may shut down to conserve power or may be limited in data-processing power or available bandwidth. These and other reliability questions have to be studied with a special care if distributed embedded systems of the future are to be trusted.

Moreover, some distributed embedded systems may operate unattended and be used to control hazardous devices or systems, which through either normal or flawed operation could lead to significant human, economic, or mission losses. Unfortunately, similar problems were encountered earlier in manufacturing automation [5, 10]. But now modern systems are potentially larger, more distributed for sure, and operate in much less controlled environments. The constraints cast on distributed embedded systems, including long life time periods, changes in structural parts, and resource limitations tend to strain existing methods for evaluating and ensuring system safety.

Unfortunately, accidents related to software already are starting to increase in proportion to the growing use of software to control potentially dangerous systems. Networking embedded systems together, as envisioned for many new applications, will only add to these problems by enabling a larger number of potentially more complex interactions among components--interactions that cannot be anticipated or properly addressed by system users. This results in a fact that fresh system designs and software engineering frameworks are needed to deal with these problems and enhance the safety of distributed embedded systems.

So, the *safety* refers to the ability of a system to operate for a reasonable period of time (within the constraints of its actual life time or MTTF {Mean Time To Failure}) without causing an accident or an unacceptable loss. Numerous distributed embedded systems will not present significant safety problems even if they fail, although such failures might frustrate or inconvenience users. However, factory floor system failures may raise significant safety issues. In fact, safety and reliability do not necessarily are foreseen going hand in hand. Thus, an unreliable system or its subsystem is not necessarily unsafe (actually, it may always fail into a safe state or an erroneous software output may not cause the system to enter an unsafe state, or a system that stops working may even decrease safety risks), while a highly reliable system may be unsafe (as the specified behavior may be unsafe or incomplete, or the system may perform unintended or unspecified functions). Thus, the simple increase of the reliability of the software or system may have no effect on safety and, in some cases, may actually reduce safety. Reliability is defined in terms of conformance with a specification, while accidents usually result from incorrect specifications.

Distributed embedded system implementation result in additional difficulties to the process. These systems greatly increase the number of states and behaviors that must be considered by the new design and the complexity of the interactions among potentially large numbers of interconnected components. While all large digital systems experience similar problems, distributed embedded systems are unusual in that many operate in real time and with limited direct human intervention. This results in a fact that these are often either unattended or managed by human operators who lack technical skills or is untrained. Furthermore, distributed embedded systems afford the possibility of more dynamic configuration than do many other types of systems. Numerous distributed embedded systems are a subject to arise from extensions for specific purposes of existing systems or from several systems connected together or related to each other in ways unanticipated by the original designers.

Next, safety must be designed into the system, including the HMI (Human-Machine Interface) and interaction. Thus, new design techniques will be required to enforce adherence to the constraints of safety of the system in distributed embedded system acting scheme and eliminate (if unfortunate, minimize) critical operator errors. Additionally, designers often make claims about the independence of components and their failure modes to simplify the design process and make systems more amenable to analysis.

Unfortunately, they usually lack adequate tools and methodologies for ensuring independence or generating alerts about unknown interdependencies. In fact, the system itself, or the design tools, will need to provide support for such capabilities. This may well require changes in the way computer scientists approach these sorts of problems as well as collaboration with and learning from others, such as systems engineers, who have addressed these issues in different domains. The scarcity in existing hazard analysis techniques when applied to distributed embedded system need to be identified.

The majority of the accidents, which are related to software suffer from requirement flaws, because incorrect assumptions about the required behavior of the software, and the operational environment. In most accidents involving systems controlled by computer, the software performed according to specification but the specified behavior was unsafe. In general, improved specification and analysis techniques are needed in this case to deal with the challenges posed by distributed embedded systems. These techniques should take into account that user needs and therefore specifications will evolve.

As a solution in regulated industries (also even in unregulated ones) in which liability or costly recalls are a concern, special procedures are required to provide evidence that fielded systems will exhibit adequate levels of safety. An implementation of distributed embedded systems greatly complicates the activities performed for such assurance, and new approaches are needed while the complexity and potential number and variety of potential failure modes or hazardous system behaviors increase.

The next consideration is *security*, which can be difficult to achieve universally, in information systems of all types, but

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf

will perhaps be especially hard in distributed embedded systems. In fact, the deployment of distributed embedded system containing various sensor technologies allows the physical world to become more tightly interconnected with the virtual world. Moreover, the network enabling of embedded computers also tends to increase the vulnerability of these systems by expanding the number of possible points of failure, tampering, or attack, thus resulting in making security analysis more difficult as well.

The range of products into which processing and networking capabilities may be embedded greatly expands the number of nodes at which security will need to be explicitly considered and influence the expectations at each node. Numerous nodes of the upper mentioned tend to consist of presumably ordinary everyday devices in which security is not currently a concern: thermostats, audio equipment, and similar.

However, mischief will become an increasing risk factor, because a close connection to the physical world and interconnection with larger networks accessible by more people with unknown motives will make lapses of security potentially more damaging, in these systems, increasing the risks associated with the integration of distributed embedded systems. Speaking of a military context as the most demanding to such factors, of course, the compromise of even fairly common devices (such as food storage equipment or asset monitoring systems) that are part of a larger distributed embedded systems could have serious security implications.

The configurations of distributed embedded systems are much more dynamic, even fluid, than typical networked systems. The models of the operators of the distributed embedded system may be quite different from those in traditional networks. The properties analyzed have significant impact on security and privacy of the communications. For example, as an object moves from place to place, its personal area network may diffuse into other networks, such as might happen, again, in military conditions, specifically, in a battle space environment. Activity between subsystems may not be under an individual's direct control, and the individual may not understand the nature of the interactivity. Various nodes will engage in discovery protocols with entities in contexts they have never encountered before.

Some distributed embedded systems may be homogeneous and their connectivity with other networks may be straightforward. In such cases, traditional network security techniques will suffice, with policy and protection methods executing in a gateway device.

In heterogeneous, diffuse, fluid networks, traditional network security methods will not be effective. Instead of that, trust management and security policies and methods should be responsible for individual nodes and applications. This may put demands on the operating system that runs on those individual nodes. They may need to distinguish between secure operating modes and more permissive modes, especially in process of discovery, configuration, and update procedures. Although cryptographic techniques enable engineers to build arbitrarily *secure* system subsystems, assembling such elements into secure systems is a great challenge, and the computing research community does not yet understand the principles or possess the fundamental knowledge necessary to build secure systems of the magnitude necessitated by distributed embedded systems. It is significantly important to ensure that security issues are addressed at the outset of system design, so that notions of network isolation can be dealt with in a straightforward manner.

However, from the early beginning, networks are designed and often deployed *before* security issues are addressed. That sort of approach will result in problems with probably most of distributed embedded systems. The system is usually much too complex to even analyze from a security perspective in case if security design is an afterthought or a security hazard has already produced consequences.

It appears like that systems, whose ability to evolve is already hard to predict are be deployed without a full understanding of the security implications at present moment. This fact results in a suggestion that both the need to accelerate relevant research and the need for coping and compensating strategies are a subject of additional investigation. Access controls need to be devised that will be easily understood, able to protect the wide variety of information that may be collected under widely varying and often unforeseeable circumstances, and perhaps even self-configuring.

The approaches that preserve the inherent capacity to communicate over a distributed embedded system yet effectively defend against denial-of-service attacks should be investigated with special care. Security in the face of energy scarcity is a significant challenge. Also, new authentication and data integrity mechanisms that require less communication overhead are required. It may be possible to exploit heterogeneity and asymmetry within the network to allow smaller system elements to do less than larger ones. Furthermore, it may be possible to exploit the redundant components in order to detect outliers and possibly sabotaged nodes when there is redundancy in the distributed embedded system.

II. General Definitions

In this chapter, the list of general attributes and rules for the model is defined. The list consists of clear theses, where each next thesis follows the essence of the previous one.

- All the case studies involve *distributed embedded systems*, which consist of **nodes** (subsystems). Here, communication links (contiguous or intermittent) are created between nodes and data exchange take place.
- In an abstract meaning, a **node** is defined as an any set of equipment designated for gathering (acquiring), processing and visualization of data, and, most important in the problem domain, communication equipment, which allows to connect such remote node with other nodes belonging to the overall distributed system. In this case, the designation and operational characteristics of the measurement and processing equipment are not taken in an account and do not influence the course of calculations. It was defined that only the main

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf

communication equipment will be mentioned as a node in the fore coming calculations and analysis [1].

- It is defined that each node is enforcing the ability of wired (connected) equipment to speak (communicate) with other remote equipment in the environment of the distributed embedded system.
- Each **technology** [3, 4](see chapter 3) taken into account in the research has a defined set of **case studies**, which are investigated and estimated with the help of factors (see chapter 4), of which importance (and relevance) is based on **coefficients** (see chapter 4).
- Each **factor** (high-level definitions in [2-4]) is designed basing on coefficients (the scale of an estimation for all factors is uniform - *[0..9]*; that is made for an opportunity of construction of a universal mathematical model.)
- Here, each factor corresponds to the set of coefficients (sets are introduced in chapter 4) for creation of an analytical model.
- The more coefficients of a factor are equal to 0 (or maximally close, aspiring to 0), the higher is the probability of exception of a factor in each researched case study.
- *A statement, which has no direct relation to the end model* - values of coefficients directly depend on conditions (environment) of each case study:
  - the topology of the distributed system,
  - the general requirements to the system and its functionality,
  - the customer's requirements to the system and to its relative parameters (for example: cost of the equipment, charges for communication services.)

III. A List of Analyzed Technologies

Here, it is specified that each investigated **technology** has a set of case studies. Measurements of these case studies result in values for coefficients for each factor.

The relativeness (an estimation of importance or relevance) of each factor is estimated and achieved by analyzing the values of measured attributes.

The list of technologies with corresponding case studies is presented in Tables 1 and 2. In the tables situated below, each column presents a title of the technology followed by its use cases.

TABLE I
A List of Analyzed Technologies

Wired Networks	WLAN	Bluetooth
RS-232	Point-to-Point	Point-to-Multipoint
RS-422	Point-to-Multipoint	Hub-to-Hub
RS-485	Peer-to-Peer	
CAN		
DeviceNet		
Modbus		
Profibus		
Foundation Fieldbus		

TABLE II
A List of Analyzed Technologies (Continued)

GSM	GSM/GPRS	Radio Modems
GSM DATA	GPRS	Transparent
GSM DATA-HSCSD	GPRS (corporate)	Peer-to-peer
GSM SMS	GPRS SMS	Multi-repeater

IV. A List of Analysis Factors of Case Studies

The following chapter introduces a preliminary list of analysis factors for case studies implementing the given technologies. Later on, the list is enriched with the coefficients, which form each factor of the end model. The list was selected by a filtering method

- A general term "*reliability***"** can not be referenced as a common factor for the analysis model, for it is a complex-compound concept [11, 12, 13]. Thus, here we speak of a compound factor that specifies the stability of the communication links, the guarantee of delivery of data, error-correction possibilities, etc. of the system implementing the selected technology. This includes several variables (coefficients): a probability of a successful data transfer: $(1-p)^n$, expected loss of data packets: $k-k*(1-p)^n$ and similar.
- **Availability** is a factor that specifies several crucial characteristics of the system, including fault tolerance, performance and similar, including integrity and privacy of communication links
- **Adaptability** is a factor that specifies the possibility to modify or change (also improve) the configuration of the distributed subsystem network according to the new demands of the system.
- **Scalability** is a factor that specifies the possibility to enlarge the quantity of subsystem in the distributed system without significant changes of the system structure, configuration or additional expenses.
- **Complexity** is a factor that specifies the overall complexity of the system implementing the selected technology: how hard is to implement each layer of technology, including middleware communicating devices and similar.
- **Cost** is a factor that specifies the cost of implementation of the system using the selected technology, which consists of two main components: installation costs, including all the hardware, software and middleware, and running costs (per-message, per-megabyte, per-minute) if applicable.
- **Range** is a factor that specifies how far can a distributed system span in space, if it is based on the selected technology. This factor is working with the natural ranges of service for the devices implementing the selected technology.

V. A List of Measurements – Coeffiecients for the Analytical Model

In the main research, there is a set of various situations (case studies) defined for each of the given technologies. The measurements are performed in these case studies.

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf

Measurements are involved in the analytical model as influencing factors. Each measurement is not obligatory to be represented in exact numbers, having these replaced with their corresponding value - an equivalent on a scale [0..9].

The maximal and minimal value is applied on each type of "measurement", where 0 is always "an impossibility of performance" (aspires to zero), and 9 is always "unlimited opportunities" (aspires to infinity). Other entered values *[1..8]* correspond to the exact numerical values of measurements broken into 8 phases.

- **Availability** [14-17]
  - *Fault-Tolerance*: the quality of the transferred data remains within the limits of norm even with failures of a communication link
  - *High or continuous availability*: an opportunity of restoration of connection after failures at the absence of necessity of intervention of a master-repairman
  - *Performance*: provides the desirable ready response
  - *Recoverability*: can restart (resend) unsuccessfully sent portions of data
  - *Consistency*: an opportunity of automatic co-ordination of actions between several units that allows them to operate as a single entity
  - *Privacy*: an opportunity of protection of the identity, dislocation of participants of communication sessions from external undesirable nodes
- **Adaptability** [14, 15, 17]
  - An opportunity of redeploying the nodes in space (dependence of quality and an opportunity of data transmission in overall after redeployment)
  - A necessity of additional adjustment of the node in case of redeployment
  - An opportunity of changing of topology or configuration of the network constructed on given technology
- **Scalability** [14, 15, 17, 18]
  - An opportunity of addition of new nodes without the need of serious interventions of the system administrator and/or serious charges
  - An opportunity of addition of the whole new subsystems consisting of numerous nodes without the need of serious intervention of the system administrator and/or serious charges
- **Complexity** [15, 18]
  - A degree of complexity of addition of new nodes (or groups of nodes)
  - A degree of complexity of adjustment of communication links between the nodes
  - A degree of complexity of installation of the communication equipment of on each node
  - A degree of complexity of installation of all equipment of the node/nodes necessary for building the communication links (modems, antennae, amplifiers, repeaters, etc.)
- **Costs** [15, 17, 18, 19]
  - Charges on implementation of the given technology in a context of a considered case study:
    - charges on installation of the necessary communication equipment (direction of antennae, search of an appropriate place without obstacles)
    - charges on the software
    - charges on the communication equipment
    - charges on adjustment of the equipment

  Charges on the maintenance (support) of the system:
    - periodic (monthly, annual, etc.) payments for the used data link (lease of the line)
    - expenses for data transmission (constant charges on data packages, time on-line charge or for charges for the volume of the transferred data)
- **Range** [15, 17, 19]
  - The maximal distance between the nodes of the system implementing given technology, allowing to work in a nominal mode, without interference (after characteristics of the implemented technology)
  - The maximal distance between the nodes of the system implementing given technology, allowing to work in a nominal mode, without interference (in conditions of a current case study: interferences and other adverse conditions)
  - A *collateral coefficient*: distance (spatial borders) on which it is possible to redeploy the node without changes in the configuration of the system
- **Speed** [15, 17, 19]
  - The maximal throughput (in bits per second) of the communication link implementing given technology between the nodes of the system, allowing to work in a nominal mode, without interference (after characteristics of the implemented technology)

VI. The Principle of the Algorithm

For a qualitative expression of the degree of similarity of the client inquiry and the adequate technology, we used the square of Euclidean space.

The Euclidean space is the *geometric distance* in a multidimensional space and is calculated as follows:

$$(x,y) = \{\Sigma_i (x_i - y_i)^2\}^{1/2}$$ [11-13].

The square of Euclidean space is calculated by the initial data, instead of standardized ones.

We used the weight coefficient (0-1), allowing us to lower the contribution into the error of one or several parameters

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf

defined by the client. In other words, the client has a possibility to choose a series of inessential parameters and endow it with the weight coefficient (up to the complete exclusion of the factor): 1 – the parameter is accounted for 100% and 0 – the parameter is disregarded. By default, the weight coefficients are equal to unity.

The data are given in absolute magnitudes and percentage of the squared maximum possible distance.

The maximum possible squared distance between the technology and the inquiry is:

$$(x,y)_{max} = \{N * (9 - 0)^2\},$$

where N is the number of estimated parameters. In our case (seven technologies),

$$(x,y)_{max} = 7 * 81 = 567.$$

The percent of the maximum possible squared distance is:

$$((x,y) / 567) * 100.$$

The method described in this paper allows the calculated data are shown graphically in the form of histograms:

- the squares of Euclidean spaces (not exceeding the value specified by the client),
- percentage of the maximum possible squared distance (not exceeding the value specified by the client), and
- absolute values of the squared distance according to all the technologies.

VII. The Structure of Calculations

The preliminary calculations based on [4] are performed in Microsoft Excel.

Each technology was evaluated by 7 parameters: Availability, Adaptability, Scalability, Complexity, Costs, Range, and Speed [4].

The evaluation was carried out by a 10-grade scale (0-9), where 0 and 9 are the lower and upper levels, respectively. In the cases where it was impossible to estimate the technology by the parameter, it was estimated by an expert method (in continuation of [4]).

The estimates of all the technologies are visualized in the table.

The client inquiry contains the combination of estimates according to all the parameters.

If necessary, the client can introduce the weight coefficient for one or several parameters (by default, all the weight coefficients are equal to unity).

The error is calculated automatically, after the introduction of the parameters and the weight coefficients specified by the client.

Further, the correspondence between the parameters of the technology and the parameters of the inquiry is calculated. The parameters of the inquiry are subtracted from the parameter of the technology, and the resulting value is squared, $R=(x_i - y_i)^2$. This procedure is carried out for all seven parameters for each technology.

The resulting value (the squared difference between the technology and inquiry parameters) is multiplied by the weight coefficient of the respective parameter:

$$R_v = v_i * (x_i - y_i)^2.$$

The products of the squared differences and weight coefficients are summed up separately for each technology. This is exactly the measure of discrepancy between the inquiry and particular technology, or, in other words, the square of the Euclidean space in a multidimensional space of parameters [15]. The dimensionality of the space is equal to the number of parameters (seven in our case) [4]. To each inquiry and technology, in a 7-dimensional space, there corresponds a separate point in the space. The square of Euclidean space yields the measure of the similarity between the inquiry and each technology:

$$T = \Sigma_i \{v_i * (x_i - y_i)^2\}.$$

The error for an *i-th* technology T_i (the sum of squares of Euclidean spaces multiplied by weight coefficients) is deduced in two forms: absolute value (T_i) and a percent of the maximum possible error ($T_{max} = 567$; $T\% = (T_i / 567) * 100$).

The client chooses the admissible error T_d between the technology and inquiry. Only the values not exceeding the mentioned admissible error T_d are displayed in the corresponding cells. At $T_d \geq T_i$, the absolute and percent value of the error is displayed; at $T_d < T_i$, the code "no" meaning the discrepancy between the *i-th* technology and inquiry (with a current admissible error T_i).

The use of weight coefficients and/or admissible error can transform the position of the technology in the 7-dimensional space from the point to a region. In this case, we take into account all the totality of solutions (technologies [15, 19]) adequate for the inquiry within the limits of the admissible error T_i.

Thus, we realized a flexible system of calculating the error, which takes into account the importance of parameters from the viewpoint of the client.

We can also reduce the contribution to the error by describing the secondary (for the client) parameters.

The secondary parameters are described by assigning them the weight coefficients v_i.

The weight coefficients in the range $0 \leq v_i \leq 1$ are assigned to the parameters at any step (for example, 0.1, 0.01, 0.001, 0.0001, etc.) and in any combination (for example, 0.9, 1, 1, ….1, 0.3, 0.3).

The error can also be understood alternatively. In this case, the error is the use of the technology not corresponding to the inquiry. Or, more literally, the improper selection of the respective technology.

VIII. The Results

Here, we will overview the graphical representation of the results, which contains three histograms.

The final form of the inquiry of the client based on his target use case is depicted in Table III.

TABLE III
The Form of the Client Inquiry

	Availability	Adaptability	Scalability	Complexity	Costs	Range	Speed
Inquiry:	3	2	2	3	2	5	2
Weight:	1	1	1	1	1	1	1

The **weight** coefficient determines whether the desired factor is irrelevant or not. This approach allows to quickly filter the results of calculations by "switching off" one or

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf

more factors and seeing if the result is still in the acceptable range.

The diagrams are constructed automatically upon the change of the inquiry, weight coefficients v_i, or the admissible error T_d, and the new diagrams are generated. This makes it possible to choose the most appropriate technology and to model the merits and demerits of different technologies, varying them within the inquiry.

A. The Admissible Absolute Values of the Error

The admissible absolute values of the error (for technologies with $T_d \geq T_i$), i.e., the errors do not exceed the mentioned admissible error T_d. At $T_d < T_i$, the code of discrepancy to particular technology "no" is not displayed on the histogram.

The results of the calculations are shown in Fig. 1 (error is set to 5, corresponding percent error is calculated as 0.881834).

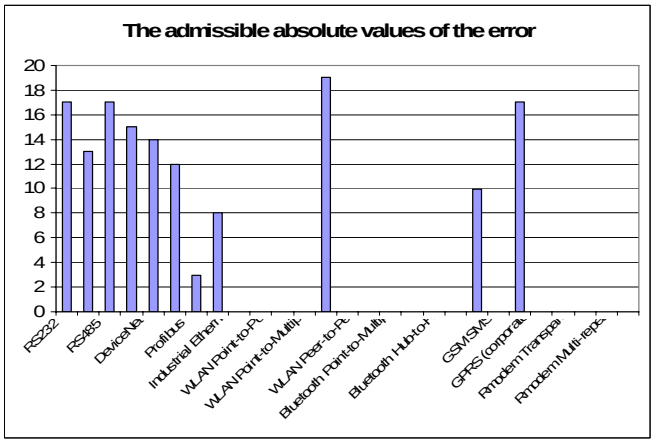


Fig. 1 The admissible absolute values of the error (5/0.881834)

It is possible to change the absolute error to a higher value for this particular use case to see the different results. The results of the calculations are shown in Fig. 2 (error is set to 25, corresponding percent error is calculated as 3.527337).

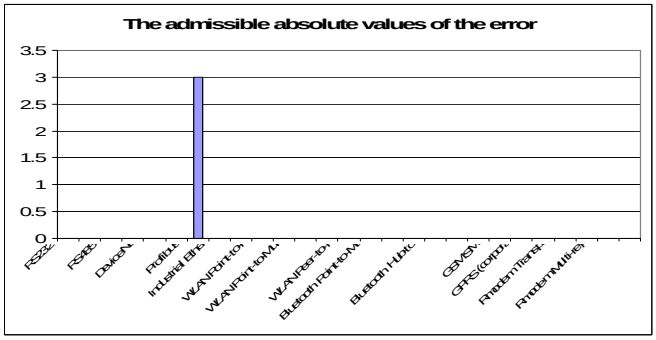


Fig. 2 The admissible absolute values of the error (25/3.527337)

B. The Admissible Percent Values of the Error

The admissible percent values of the error for technologies with $T_d \geq T_i$, i.e., the errors do not exceed the mentioned admissible error T_d. The percentage error is the same as before: ((x,y) / 567) * 100. At $T_d < T_i$, the code of discrepancy to particular technology "no" is not displayed on the histogram.

The results of the calculations are shown in Fig. 3 (error is set to 5, corresponding percent error is calculated as 0.881834).

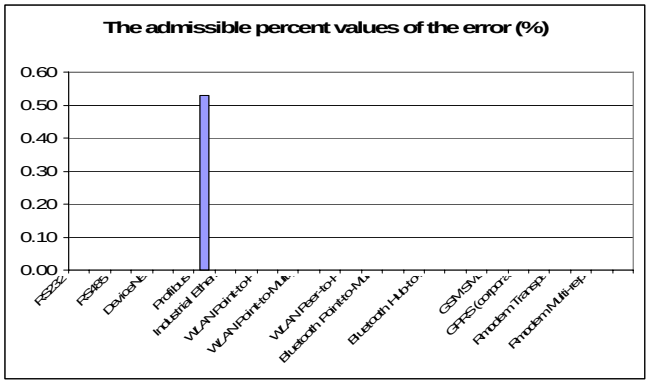


Fig. 3 The admissible percent values of the error (5/0.881834)

It is possible to change the absolute error to a higher value for this particular use case to see the different results. The results of the calculations are shown in Figure 4 (error is set to 25, corresponding percent error is calculated as 3.527337).

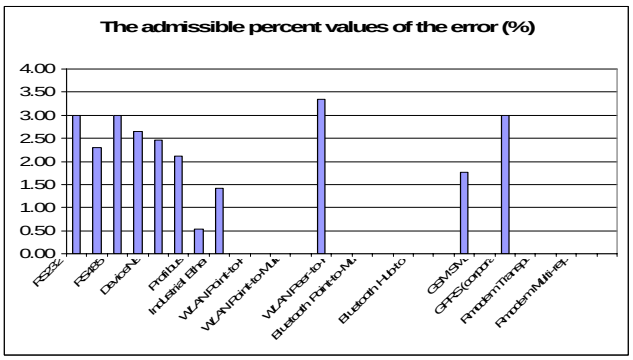


Fig. 4 The admissible percent values of the error (25/3.527337)

C. The Admissible Percent Errors

The admissible percent errors according to all the technologies is the same as before: ((x,y) / 567) * 100. The values of the admissible error T_d specified by the client are not taken into account.

The results of the calculations are shown in Fig. 5.

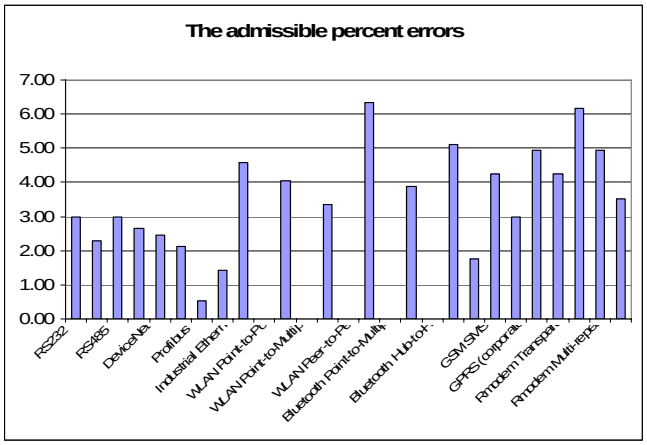


Fig. 5 The admissible percent values of the error

IX. Conclusion

The main research of the author is comparative analysis of communication technologies in industrial automation. The research results provided in the paper introduce a view on the problem domain for the main research. With this work completed, it is possible to proceed with the enhancements of the introduced model. The current model is flexible enough to provide an engineer with an ability not only to compare different technologies in a given use case, but also exclude the desired factors from the calculation, thus evaluating the impact of each of the factors (or a group of factors) on the final result.

The method introduced in this paper is an important intermediate stage in the global research for author's PhD thesis. This stage defines the base of the analytical model and provides an opportunity to proceed to the evaluation of the empirical data already acquired from numerous experiments. The current research result has helped in choosing the right solutions for locally developed embedded distributed systems in test target use cases.

References

[1] Mikoyelov D. "An Implementation of Modern Communication Models in Process Automation" // 40th Spring International Conference on Modelling and Simulation of Systems (MOSIS'06), Prerov, Czech Republic, 2006.

[2] Rusakov P, Mikoyelov D. "Subsystem Communication in a Distributed Embedded System" // 13th International Conference on Information Systems Development, Lithuania, 2005.

[3] Rusakov P, Mikoyelov D. "An Analysis of Distributed Embedded Systems in Industrial Process Automation" // Design, Analysis, and Simulation of Distributed Systems 2006, Alabama, USA, 2006.

[4] Rusakov P, Mikoyelov D. „An Analysis of Communication Technologies for Distributed Embedded Systems in Industrial Process Automation" // Proceedings of 15th International Conference on Information Systems Development (ISD 2006), Budapest, Hungary, 2006.

[5] Половко А.М. and Гуров С.В. "Основы Теории Надежности: Практикум" // БХВ-Петербург, Санкт-Петербург, 2006 [in Russian].

[6] Рутковская Д., Пилиньский М., Рутковский Л. «Нейронные сети, генетические алгоритмы и нечеткие системы» // Горячая Линия – Телеком, Москва, 2004 [in Russian].

[7] Зубанов Н.В. «Анализ устойчивости относительно поставленной цели как один из подходов к описанию функционирования организации в условиях неопределенности» // Монография, Самара, 2001 [in Russian].

[8] Uchida R., Okada H., et al. "Influence of Wireless Transmission Performance on Quality of Wireless Control" // Nagoya University, 2005.

[9] Лойко В.И., "Методика Системного Анализа Прикладных Процессов Акустомагнитной Обработки Жидкости" // Научный электронный журнал КубГАУ, No 01(9), 2005 [in Russian].

[10] Zhou Zhongding Z., Xiongjian L. "A Data Analysis Model of Reliability on Communication Networks Based on Discrete Fourier Transform" // Beijing University of Posts and Telecommunications, 2003.

[11] Кремер Н. Ш., "Теория вероятностей и математическая статистика" // Юнити-Дана, 2004 [in Russian].

[12] Гмурман В.Е., "Теория вероятностей и математическая статистика: Учебное пособие для вузов Изд.12-е, перераб." // Высшее образование, 2006 [in Russian].

[13] Гмурман В.Е., "Руководство к решению задач по теории вероятностей и математической статистике: Учебное пособие для вузов Изд.12-е, перераб." // Высшее образование, 2006 [in Russian].

[14] Wenliang Du, Jing Deng, et al, "A pairwise key predistribution scheme for wireless sensor networks" // ACM Transactions on Information and System Security (TISSEC), 2005.

[15] Ian F. Akyildiz, Xudong Wang, et al, "Wireless mesh networks: a survey" // Computer Networks, Volume 47, Issue 4, (2005).

[16] Karlof, C. Wagner, D., "Secure routing in wireless sensor networks: attacks and countermeasures" // Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE, 2003.

[17] Marcos A.M.V., Diógenes C.S.J., "Survey on Wireless Sensor Network Devices" // 2nd ACM International Workshop on Wireless Sensor Networks and Applications, 2003.

[18] Lakshmi V., "Design Trade-offs in Wireless Sensor Network System Development" // Research and Technology Center, Robert Bosch Corporation, 2004.

[19] Kumar K.D., Karunamoorthy L., "An Infrastructure for Integrated Automation System Implementation" // Springer Netherlands, International Journal of Flexible Manufacturing Systems, 2004.

Dmitry A. Mikoyelov received the Bachelor's and Master's degrees (both in Applied Computer Science) from the Riga Technical University (RTU), Riga, Latvia, in 2000 and 2003, respectively. The major field of study is analysis of communication technologies in factory automation.

He is a PhD student in the Department of Applied Computer Science at the Riga Technical University, Riga, Latvia. He is a Development Group Leader in FMS Software, a Latvian company working in the field of financial management systems. He also conducts advanced training courses in the development of automation systems and project management in the field of factory automation.

Mg.Sc.Ing. Mikoyelov received an "excellent science advisor" honor diploma in 2006; successfully finished a Wonderware's International Application Consultant course in 2006.

Open Science Index, Computer and Information Engineering Vol:1, No:5, 2007 publications.waset.org/8165/pdf