Quality of Automation

Wolfgang A. Halang
Chair of Computer Engineering, FernUniversität, Hagen

Abstract: As manufacturing and logistics processes have a significant impact on the quality of products and services, the quality of the automation involved should be considered as well. To this end, a list of quality criteria for automating systems is proposed, consisting of qualitative exclusive, qualitative gradual and of quantitative criteria. It is shown that qualitative characteristics are much more important than quantitative ones. Provided that they fulfill the exclusive criteria, automating systems may then be compared on the basis of the gradual and quantitative ones. Since quality of automation also depends on the quality of the corresponding engineering processes, some criteria are compiled to evaluate the quality of automating, and critical points of technological regress in the design and development processes of automation systems are identified. Finally, the quality of automation as such is addressed and several requirements for it mentioned. It is indicated that automation may not always be useful, and that it can turn out harmful to society.

Keywords: automation, automating, automating system, quality, quality criteria, technological regress

1. Introduction

Whereas the quality of products and services produced by automated systems receives considerable attention, this seems not to be the case for automation itself and the lifecycle of automating systems including conceptual phase, design, construction, deployment and operation. Thus, it may be timely to have a look into this issue.

Although widely used in everyday life and intuitively evident, first the meaning of the term "quality" needs to be clarified. Looking for definitions of quality, one finds quite many of them, and they all appear rather vague. For automation engineers it is appropriate to turn to a definition given in the international standard ISO 9000 dealing with quality issues. There it reads [5]:

> *The quality of something can be determined by comparing a set of inherent characteristics with a set of requirements. If those inherent characteristics meet all requirements, high or excellent quality is achieved. If those characteristics do not meet all requirements, a low or poor level of quality is achieved.*
>
> *Quality is, therefore, a question of degree. As a result, the central quality question is: How well does this set of inherent characteristics comply with this set of requirements? In short, the quality of something depends on a set of inherent characteristics and a set of requirements, and how well the former complies with the latter.*
>
> *According to this definition, quality is a relative concept. By linking quality to requirements, ISO 9000 argues that the quality of something cannot be established in a vacuum. Quality is always relative to a set of requirements.*

Hence, according to this definition, it is necessary to establish sets of requirements if we want to assess the quality of automating systems and of automation as such. It is the objective of this contribution to give it a try.

In the next section, we commence with compiling some requirements automating systems are expected to meet. Automating systems are the ones that control the operation of automated processes or systems of any kind ("systems/equipment under control"). Nowadays, the automating systems are mainly computerised, and the control computers installed ("embedded systems") predominantly operate in the real-time mode. Owing to this, the author derives his arguments from a book dealing with quality of service of real-time computing [3]. Whereas the considerations on the quality of automating systems, i.e. technical artifacts, will be relatively systematic, the quality of automating, i.e. the proper work carried out by automation or control engineers, will be treated rather anecdotically by giving a few thought-provoking impulses. It is hoped that they will lead to a more thorough elaboration of these issues by further authors.

2. Automating Systems

With the increasing dependence of society on automation systems, the concern about their performance and safety is growing. When deciding on the selection of appropriate components and systems, one has to be aware of their benefits and hazards for the given application and make corresponding choices. The decision criteria differ from application to application. They include the following:

- suitability for the application,
- dependability (robustness, correctness),
- safety,
- security,
- reliability,
- availability,
- user friendliness (ergonomics),

– maintainability,
– flexibility,
– performance,
– efficiency,

. Automation applications impose many requirements on the different parts of automating systems. A fundamental one is the ability to keep temporarily pace with the systems being automated, i.e. to operate in synchrony with them and their environments. Automating systems must fulfill this

Timeliness

requirement even under extreme load conditions. When requested by the automated processes, sensing, evaluation and appropriate actuations must be performed on time. Not the mere processing speed is decisive for this, but the timeliness of reactions within pre-defined and predictable time bounds. Hence, it is characteristic for automating systems that their functional correctness does not only depend on the processing results, but also upon the instants when these results become available. Correct instants are determined by the environments, which cannot be forced to yield to automating systems' working speed. The timing constraints typically result from the physical laws governing the processes being automated.

The dependability requirement implies permanent availability, which can only be provided by fault-tolerant and – especially with respect to inadequate handling – robust systems. Naturally, expecting high dependability does not mean the unrealistic demand that systems never fail, since no technical system is absolutely reliable. Using appropriate measures, however, one must strive to make malfunction quantifiable and as unlikely as possible. In doing so, the individual limitations of automating systems must be recognised, and the risk of their utilisation in process control must be pondered carefully.

Another fundamental requirement automating systems need to fulfill is full behavioural

Predictability

as their actions and reactions carried out must be precisely planned before deployment and fully predictable at any time of operation. This particularly holds for the case when several events occur simultaneously, leading to a situation of competition for service, and also includes transient overload and other error situations. Then, systems are expected to degrade their performance gracefully in transparent and foreseeable ways. Only fully deterministic system behaviour will ultimately enable the safety licensing of automating systems for safety-critical applications. Predictability supplements the timeliness demand, as the latter can only be guaranteed if a system's execution behaviour is precisely predictable, both in time and with respect to external event reactions.

Predictability is a degree of trust one is justified to have that a correct forecast of a system's state can be made qualitatively or quantitatively. It represents a degree of behavioural determinism even in error situations. Functional and temporal predictability are considered here as one integral property. When discussing a system's behavioural predictability it is reasoned upon the portion of cases in which the system behaves as predicted or, when comparing two systems, which one of them behaves predictably in more cases. Here mostly timeliness in connection with fault forecasting and prevention measures applied in order to achieve better functional and temporal predictability are concerned.

Similar statements as above apply also to dependability. While predictability is mainly addressed by fault forecasting and fault prevention, dependability on the other hand is supported by fault tolerance and fault removal mechanisms. Dependability is a degree of trustworthiness of an automating system, which allows reliance to be justifiably placed on the service – function – it delivers. According to [1], it is considered as the "sum" of availability, reliability, safety, security, integrity and maintainability. Depending on the application, one might ponder the individual "summands" with respect to their relative importance to the concrete application.

There are many questions to be asked when discussing the quality of an automating system. First of all, it needs to be inquired whether a control method selected is suitable to solve the given automation problem. If so, the question arises whether the system's execution behaviour is (fully) predictable, i.e. known in all situations. If the answer is yes, then this obviously contributes to the system's quality. If the answer is no, the system may be disqualified. Then, for example, it may be inquired whether the system considered is fault-tolerant. If not, the question remains as to whether the system endangers its environment upon breaking down. If the latter is the case, then the system's quality is obviously insufficient and it should not be used.

Some other questions that follow from the above strategy are: Does the system degrade its performance gracefully upon occurrence of malfunctions, and does it shut down in a safe and controlled way? Can the system be re-configured or updated during operation? If the answer is no, and the system may never shut down after start-up, e.g. in a space station, then the system is clearly not usable for continuous operation. Do situations exist in which the system cannot keep pace with the automated process? If so, the system must be disqualified. Is the system's worst-case performance known? Is it accounted for? Does the system perform correctly, i.e. does it conform to the given functional specification?

These and many other questions give rise to a number of appropriate requirements, i.e. quality criteria, which can be divided into three groups:

Qualitative exclusive: criteria either being fulfilled or not

Qualitative gradual: one system may have a property to a higher degree than another one, but the property cannot be quantified precisely

Quantitative: measurable criteria

The assessment of qualitative criteria results in questionnaires and decision trees. It is important to note that the criteria cannot be used in a general setting but, consistent

with the definition of quality from ISO 9000, *only in an application-specific way*. The quality of an automating system needs to be assessed by a holistic examination. The quality lies in the extent to which the system fulfills the automation engineers' expectations. The following criteria were identified as appropriate and then properly grouped:

1) Qualitative exclusive criteria:
   - timeliness – the ability to keep pace with the automated process,
   - functional correctness,
   - fail-safe operation,
   - permanent readiness – non-stop timely availability of correct service,
   - all applicable physical constraints met,
   - licensability (for safety) – certification.
2) Qualitative gradual criteria:
   - suitability – the degree of the employed automation/control method to be suitable for the application,
   - timeliness,
   - availability – the proportion of time a system is in a functioning condition,
   - reliability Ő- the degree of a system's ability to perform its required functions under stated conditions for a specified period of time,
   - safety – the degree of protection against risks caused by errors,
   - security – the degree of protection against consequences of failures being induced from a system's environment,
   - integrity – the degree of absence of improper system state alterations; further being determined by:
   - robustness – the degree of a system's resilience under stress, and
   - fault tolerance,
   - complexity (simplicity),
   - maintainability – the degree of ability to undergo repairs and modification; further being determined by:
   - portability – the degree of effort needed to port a system to a new environment or platform, and
   - flexibility – the degree of a system's adaptability to a new environment.
3) Quantitative criteria:
   - timeliness,
   - execution time,
   - worst-case response times to occurring events,
   - worst-case times to detect and correct errors,
   - Mean Time Between Failures (MTBF), Mean Time To Failure (MTTF) and Mean Time To Recover (MTTR),
   - capacity reserves,
   - overall project costs.

Traditionally, quantitative measures form the most popular aspects of performance analyses. Hence, it may come as a surprise that just a few of the criteria compiled above fall into this category. Care must be taken when employing the measures MTBF, MTTF and MTTR as, for example, a mean time between failures of 100,000 hours does not exclude a break-down during that time. Criteria relating to implementation artifacts or technicalities have been omitted in favour of those that matter on the overall system level.

Being fail-safe or fail-secure is the property of a system/device that, if (or when) it fails, then it fails in such a way as not to cause harm, or only minimum harm, to other devices or danger to personnel. Fail-safe operation means automatic protection of programs and/or processing systems when a hardware or software failure is detected in an automating system.

Availability is the proportion of time a system is in a functioning condition. It also represents readiness for correct service or the degree to which a system suffers degradation or interruption in its service as a consequence of failure of one or more of its components. A system's availability can be supplemented by graceful degradation. Upon malfunctions the latter represents a degree of a system's functionality provided in error situations. If the system's operating quality decreases at all, the decrease should be proportional to the failure's severity. The mechanisms to achieve this originate from the domain of fault tolerance.

Reliability is a degree of a system's ability to perform its required functions under stated conditions for a specified period of time. Reliability represents the continuity of correct service (both functionally and temporally) and depends on other dependability criteria as well. During design it is best supported by fault prevention, however during operation, fault tolerance mechanisms ensure that it is sustained. Also, during design and operation security methods and mechanisms apply to ensure secure operation.

Safety represents the degree of protection against risks caused by errors. It is closely related to reliability and robustness, which both support a system's safety. A safety-critical application represents one whose malfunction introduces risk to human lives/health or can cause damage to the system itself and/or its environment. A degree of safety can be established (also formally, e.g. in the form of Safety Integrity Levels (SIL) [4]), which may be sufficient or insufficient with regard to the safety requirements set in their specifications. For safety-related systems two kinds of requirements are defined:

1) Safety function requirements: the safety functions that have to be performed
2) Safety integrity requirements: the reliability with which the safety functions have to be performed

By functional safety the ability of a safety-related system to carry out the actions necessary to achieve a safe state for the automated system or to maintain a safe state for it is meant and relates to safety. Safety integrity is the likelihood of a safety-related system to achieve safety functions required under all conditions stated within a given period of time; it relates to reliability.

Security is a degree of protection against consequences of failures being induced from the environment of a system.

It is seen as a "sum" [1] of availability, confidentiality and integrity – in the sense of absence of unauthorised state alterations, where confidentiality represents the degree of protection against unauthorised disclosure of information. As with safety, the degree of security depends on the methods and mechanisms applied to ensure a system's secure operation.

Integrity represents the absence of improper system state alterations [1]. It is prerequisite for safety, availability and reliability, and is rarely considered as a stand-alone attribute. Although the mentioned relation does not hold for confidentiality, one could also consider security in the context of integrity. There is no direct relation between integrity and maintainability, but maintenance should not compromise a system's integrity. Integrity is supported by robustness, which is achieved by fault tolerance measures, and simplicity of a system's design. A complex system design is likely to undermine the system's integrity.

Robustness is a degree of a system's resilience under stress, or when confronted with invalid inputs or changes in internal structure or external environment. Robustness supports integrity and is closely related to reliability. In contrast to reliability it concentrates on the number of changes, fault rate, or number of deadline misses that the system "survives" intact.

Fault tolerance is the degree of a system's provided ability to continue working properly in the event that some of its components fail. It is the property that enables a system to continue operating properly upon occurrence of a failure in (or of one or more faults within) some of its components. Protective fault tolerance mechanisms are defined and implemented during the system creation phases of the life-cycle, whereas their actions are effective during the operational phase. Fault prevention and removal techniques allow one to increase reliability (reducing the probability of fault occurrences), or availability in the case of repairable systems (e.g. detect-and-repair mechanisms). Observing safety criteria leads to employ techniques related to fail-safe design in order to achieve the highest degree of dependability by integrating mechanisms that allow for full continuation of a system's mission despite the presence of faults, thus increasing its integrity. Such mechanisms may deliver full service at no reduction in performance. Due to the presence of faults, however, the performance usually decreases to an acceptable level ("graceful degradation"). Employing fault tolerance techniques has a direct positive impact on safety, reliability and availability.

One may say that (unnecessary) complexity generally diminishes integrity while, on the other hand, simplicity (indirectly) fosters it.

Maintainability represents the degree of the ability to undergo repairs and modifications. Maintainability is an integral property supported by portability and flexibility.

Portability represents the degree of effort needed to transfer a system to another environment or platform. Portability is affected by a system's complexity and the number/complexity of the components that need to be (ex-) changed. It is determined by the number of platforms on which an application can run. This criterion is carried over to adaptability in the sense of the number of changes that have to be made in order to be able to port (transfer/translate) the application to another platform.

Flexibility represents the degree of a system's adaptability to a new environment. It may also be considered as resilience in recovering from a shock or disturbance originating from (a change in/of) the environment. Similar statements as for portability also apply to flexibility, which represents a measure for the effort to achieve adaptability regarding portability and subsequent (version) upgrades. The degree of flexibility is highest for applications where portability and adaptability have been accounted for during their design phases.

The "bottom line" of all considerations is often cost, and a general optimisation objective is to minimise it. As overall cost comes about in an additive way, there is wide room to consider trade-offs and to make compromises in choosing hardware and software components under observation of the limitations specified.

3. Automating

After having considered quality criteria for the assessment of automating systems in the preceding section, now some essential requirements are to be compiled which the process of engineering them, i.e. of specifying, designing, constructing, deploying and verifying or validating them, can be expected to fulfill. Most of the criteria identified above for automating systems directly give rise to related ones useful in evaluating the quality of automating. Therefore, they will not be explicitly mentioned here again. Partly not covered by criteria from the last section are the following requirements for the engineering processes:

- Safety licensability of systems engineered
- Suitability of the automation/control methods, tools and artifacts employed
- User friendliness (simplicity) of employed tools
- Understandability of the design documentation
- Productivity at low costs

All criteria in this list are qualitative, with the first one having exclusive and the other ones gradual character. Since engineering processes are carried through by humans, the concern for human factors is of utmost importance. If the tools and artifacts employed are simple, easy to understand and to handle, as well as oriented at the engineers' way of thinking, safety is inherently fostered.

The second requirement in the list above, viz. suitability, appears to be trivial at first sight. The following examples of state-of-the-art automation engineering reveal that this is not the case.

The task is the fundamental concept to realise the asynchronous model of real-time programming. Tasks are the elements of concurrency, parallel execution and of reactivity to any kind of events. Therefore, tasks are available as language constructs in genuine real-time programming languages. In automation engineering it is, however, common practice not to use such languages, but C. As C does not know tasks, each time a task is needed, a work-around needs to be programmed and an operating system function to be invoked. Fig. 1 shows the long segment of – unintelligible – C code required to schedule periodic activations of

```
#include (stdio.h>
#include <time.h>
#include <sys/timers.h>
#include <sys/proxy.h>
#include <sys/kernel.h>
#include <signal.h>
void main()
{
pid_t timer_proxy, end_proxy, task_pid;
timer_t id, id_2, i;
struct itimerspec timer, timer_end;
struct itimercb timercb, timercb_end;
timer_proxy=qnx_proxy_attach(0,0,0,0);
if(timer_proxy==-1){
     printf("Unable to attach timer_proxy");
     return;
}
timercb.itcb_event.evt_value=timer_proxy;
id=mktimer(TIMEOFDAY,_TNOTIFY_PROXY,&timercb);
if(id==-1){printf("Unable to attach timer\n");}
timer.it_value.tv_sec=4L;
timer.it_value.tv_nsec=0L;
timer.it_interval.tv_sec=1L;
timer.it_interval.tv_nsec=0L;
reltimer(id,&timer,NULL);
end_proxy=qnx_proxy_attach(0,0,0,0);
if(end_proxy==-1){
     printf("Unable to attach end_proxy");
     return;
}
timercb_end.itcb_event.evt_value=end_proxy;
id_2=mktimer(TIMEOFDAY,_TNOTIFY_PROXY,&timercb_end);
if(id_2==-1){printf("Unable to attach timer\n");}
timer_end.it_value.tv_sec=time(NULL)+10;
timer_end.it_value.tv_nsec=0L;
timer_end.it_interval.tv_sec=0L;
timer_end.it_interval.tv_nsec=0L;
abstimer(id_2,&timer_end,NULL);
for(;;)
     {
          Receive(timer_proxy,0,0);
          spawn1(P_NOWAIT,"task1","task1",0,0,0,);
          printf("Tick\n");
          if((task_pid=Creceive(end_proxy,0,0))==end_proxy)
          {
               rmtimer(id);
               rmtimer(id_2);
               return;
          }
     }
}
```

Fig. 1. C code for task activation under QNX
Rys. 1. Kod w języku C aktywacji zadania w QNX

a task under QNX. The reader is not expected to go into the details of this code segment, but only to compare it with the following single program statement providing the same functionality:

at 12:00:00 **all** 1 SEC **until** 12:00:10
activate task1 **priority** 5;

Being very close to plain text in natural language, this statement formulated in the real-time language PEARL90 [2] expresses the functionality concisely, is self-documenting, simple and easily understandable – even to the layman. Needless to say that the usage of suitable tools, such as presented here, fosters safety of the systems designed and high productivity of the engineering process. The scandal is that the syntax of the task activation statement shown was defined already in 1969, and that practising engineers refuse to use such a tool.

Shortly after having been defined, the language PEARL and the language construct above were available for routine use on process control computers, at that time called minicomputers, which were equipped with real-time operating systems and a host of engineering tools. After some years, however, there had to be a new fashion. The process control computers were blackmailed for being centralised, although they seamlessly transform into nodes of distributed systems and their architecture scales in both directions. The new fashion was to replace them by the rather primitive and low-performing programmable logic controllers with inferior programming capabilities. It is interesting to note that the wheel was re-invented: the process control computer appeared again on the scene in form of the industrial personal computer (IPC), and the re-inventor became a celebrity in automation circles. As far as its hardware is concerned, the IPC is just a slight update of the former minicomputer. With respect to the operating systems and programming capabilities provided, however, it falls considerable short of the state reached 40 years ago.

Another example of inadequate state-of-the-art practice is the situation in field level communication. For this purpose, in the last decades a number of proprietary and of internationally standardised fieldbus systems has been developed. Particularly the standardised fieldbuses widely proved themselves in industrial applications. Although they provide deterministic real-time behaviour and some even support safety-related applications, these fieldbuses are being more and more replaced by employing Ethernet technology. The corresponding equipment is cheaper, but the costs for this advantage are lacking real-time guarantees and opening the door to the intrusion of malware. Thus, little savings in hardware cost are traded in for severe safety and security hazards whose consequences may be loss of human lives, harm to the environment and enormous liability payments.

Safety and security hazards are considerably further exacerbated by following the latest fashion, viz. wireless automation networks. First of all, wireless communication is inherently less dependable and more susceptible to disturbances than transmission over wires. The data packets transmitted can easily be intercepted and altered by unauthorised persons, thus opening the floodgates to espionage and sabotage. For industry, ostensible savings in wiring costs may, thus, turn into existential endangerments.

The examples above show clearly that automation engineers not always strive to use the most suitable tools or artifacts in designing automating systems. Making design decisions for reasons of, for instance, fashions is unprofessional. Subjecting also engineering processes to strict quality assessment may be a remedy for this unsatisfactory situation.

4. Automation as Such

When considering quality of automation, also the quality of automation as such should be addressed, i.e. how well automation complies with the requirements of those putting automation systems in place and of society in general.

Originally, the driving force to develop automation was the desire to relieve people of hard physical or stupefying routine labour as well as of work in harsh environments or under dangerous conditions. Later further demands added

to this, such as increasing speed and accuracy of production, or even to carry out functions which humans are unable to do for whatever reasons. This development has led to the present situation that practically anything is automated that can be automated in order to save the cost of human labour.

From the business management point of view, it makes sense to minimise costs and to maximise usage of resources. From a more global point of view, however, e.g. the one of national economics, certain kinds of automation can turn out not to be useful and even harmful, as the following examples will reveal.

That a guard with a revolver stands next to an automated teller machine is just ridiculous. Nothing is achieved. The overall system of ATM and guard may be even more costly than a classic teller with a bank employee. At least, the number of jobs involved is unchanged.

Ridiculously useless are also the fashionable electronic door locks. To open a door, classical mechanical keys are replaced by personalised transponders wirelessly communicating with a device in the door lock to identify persons authorised to open the door. The electronic check takes longer than to manually open a door with an old-fashioned key. And once access is approved, one still has to open the door with muscular strength. Transponders can be lost as easily as mechanical keys. The only "advantage" of this kind of automation may be that it enables employers, neighbours or secret services to observe people more easily.

There is no more personnel at small train stations. Instead, there are surveillance cameras monitored in remote control centres, from where also announcements are made via loudspeakers. Tickets must be purchased in automatic vending machines. These are rather expensive, but often fully or partly out of service, for instance, they coin checking components do not function properly. When malfunctioning, thus, these machines turn passengers willing to pay into criminals if they board trains without having tickets. This example shows clearly that automation may not be beneficial. It causes customer dissatisfaction and leads to the loss of many jobs, particularly of people with low qualifications. As they do not find other jobs, society needs to finance their unemployment. If they would be employed as salespersons for tickets, not only the expensive machines could be saved, but also repairs of the stations made necessary by vandalism, which would be prevented by the sole presence of employees.

5. Conclusion

Quality of products has always been an important consideration for any producer, and increasingly sophisticated products and processes are spurring a growth of this importance. Since production processes have a significant impact on the quality of products, the quality of the corresponding automation should be considered as well. As quality measures for automating systems have obviously not received much attention, yet, a look into their characteristics led to the proposal of a – still incomplete – list of qualitative exclusive criteria, qualitative gradual criteria and quantitative criteria. It turned out that qualitative characteristics are much more important for automating systems than quantitative ones. Provided that they fulfill the exclusive criteria, automating systems may then be compared on the basis of the gradual and quantitative ones. Since the latter mainly relate to costs, quality comparison becomes possible – always under the constraint, however, that the indispensable exclusive criteria are met. Quality of automation also depends on the quality of the corresponding engineering processes. To evaluate the quality of automating some criteria were identified, and critical points of technological regress in the design and development processes of automation systems were mentioned. Finally, the quality of automation as such was addressed and several requirements for it mentioned. It was indicated that automation may not always be useful, and it can become harmful to society.

Bibliography

1. Avizienis A., Laprie J.-C., Randell B.: *Fundamental Concepts of Dependability*, Research Report N01145, LAAS-CNRS, 2001.
2. *DIN 66253-2: Echtzeitprogrammiersprache PEARL90*, Beuth-Verlag, Berlin 1998.
3. Gumzej R., Halang W. A.: *Real-Time Systems: Quality of Service Assessment – Introducing Quality of Service Parameters into Design and Development of Real-time Systems*, Springer-Verlag, London 2010.
4. *IEC 61508: Life-cycle Management of Instrumented Protection Systems*, International Electrotechnical Commission, Geneva 1998.
5. *ISO 9000*, cited from [http://www.praxiom.com].

Jakość automatyzacji

Streszczenie: W artykule zaproponowano kryteria jakościowe (wyłączające i stopniowalne) oraz ilościowe oceny jakości systemów automatyki. Pokazano, że charakterystyki jakościowe są dużo ważniejsze od ilościowych. Wykazano, że po spełnieniu szczególnych warunków systemy automatyki mogą być porównywane ze sobą na bazie kryteriów stopniowalnych i jakościowych. Opracowano kryteria ewaluacji jakości automatyzacji oraz zidentyfikowano punkty krytyczne regresu technologicznego w projektowaniu i rozwijaniu systemów automatyki. Omówiono szereg wymagań odnośnie jakości automatyzacji *per se*. Wskazano także, że automatyzacja nie zawsze jest użyteczna, a wręcz może być społecznie szkodliwa.

Słowa kluczowe: automatyka, automatyzacja, system automatyki, jakość, kryteria jakości, regres technologiczny

Wolfgang A. Halang, Prof. Dr.

Wolfgang A. Halang received a doctorate in mathematics from Ruhr-Universität Bochum in 1976, and a second one in computer science from Universität Dortmund in 1980. Since 1992 he holds the Chair of Computer Engineering in the Faculty of Electrical and Computer Engineering at Fernuniversität in Hagen, Germany, whose dean he was from 2002 to 2006.

e-mail: wolfgang.halang@fernuni-hagen.de