NIDES as an IS Audit Tool

NIDES as an IS Audit Tool

D P Drljača[1], D Starčević[2] and B Latinović[2]

[1]Independent University Banja Luka, Veljka Mlađenovića 12E, 78000 Banja Luka, Republic of Srpska, Bosnia and Hezergovina
[2]Paneuropean University APEIRON Banja Luka, Pere Krece 13, 78000 Banja Luka, Republic of Srpska, Bosnia and Hezergovina

E-mail: drljacad@gmail.com

Abstract. The aim of this paper is to present the role of the IDS system in the process of information system audit. The paper introduces the IDS systems, their organization and architecture, and the accent is placed on IDS systems of a network type that have in their architecture an associated expert system with the task of predicting malicious activities in the system in order to take protective or protective measures in a timely manner. The paper provides basic definitions, architecture and partially describes the functioning of these systems, the usefulness and need of consulting the IDS system in the process of information system audit. When auditing the information systems, a large part refers to the security or safety aspects of the functioning of the system, and therefore IDS systems are the ideal partner in the audit process, which can provide the auditor with very important data. These data can range from recognizing the established ways of using the system, specific actions in a system that can be considered potentially harmful to the data that definitely confirm malicious actions in the system.

1. Introduction

The basic problem of information systems is the security and protection of data integrity that circulates and communicates through the observed information system, as well as the components of that system as a whole. So far, a large number of papers have been published on the subject of security and security of information systems, there are a number of research problems related to technical and technological solutions to security and safety issues. However, surely the effort of each system administrator is to prevent or to anticipate possible disturbance of the security and integrity of the system it manages.

Although protection systems have been significantly improved over the past few years, there have been frequent infiltrations into systems and a deterioration in the integrity of their data, which often result either in loss of data or in high material costs. As the most effective tools currently available to system administrators, in practice, for detecting intrusion are *Intrusion Detection Systems* (IDS). These systems are rapidly evolving in order for system administrators to have the necessary level of insight into the security of their own system and related data.

The significance of this paper is that it helps to understand the functioning of such a system for the needs of the information system audit process, especially those systems where valuable data are stored, such as banking systems, military systems or public administration systems.

2. Goals of application of expert systems in audit

The auditor should, during and after the audit process, form a reasoned opinion on the integrity, reliability, usefulness and safety of the system being analysed, as well as audit techniques that are assisted by the computer. Therefore, it is very important to take this into account when preparing the audit. The auditor must anticipate the ways of collecting, processing and testing data and supporting documentation for support. It is important that the auditor is aware of the capabilities and potential of the expert system used in the audit process in order to make sure that the decision lines are adequate and best suited to the given environment and the system that is subject to the audit.

The practice of a financial audit is that the auditor starts the audit planning with one general audit objective. The auditor should obtain a reasonable assurance that the client's financial statements have been presented in a valid and correct manner in accordance with all material aspects and in accordance with the principles of international auditing standards. In order to achieve its objective, the auditor must obtain all the necessary and sufficient records of evidence that will serve as the basis for obtaining an audit opinion.

Similar to this definition, we can also define the objectives of the application of expert systems in the audit:

The auditor must be able ***to obtain audit evidence*** *using the expert system, which is* ***recorded in both electronic and written form*** *in the accompanying support documents. In addition, the expert system must draw* ***attention to potential inconsistencies*** *that can be identified using artificial intelligence techniques (such as neural networks).*

Section404 of so-called Sarbanes-Oxley Act of 2002 (SOX) and two directives of the European Parliament and of the Council of Europe (Directives 2006/43 / EC and 2008/30 / EC of the European Parliament and of the European Council - 'EuroSOX') commit to the establishment, documentation and management of the internal control system and their subsequent audit as part of the audit risk assessment [1], [2]. This means that it is necessary to take all necessary steps to ensure that the internal control system is reliable and provide adequate and credible findings. Also, such a system must satisfy the requirements of the said legal documents, but also to operate in accordance with best practices and standards in the field of information system audit. Therefore, the selected audit objectives, which are implemented with the help of expert systems, must be harmonized with these legal requirements and solutions.

Experience has shown that accounting and systems are most sensitive to data downturns and modifications, whether human error (intentional or accidental) or abuse. It is the purpose of the audit to identify potential forms of security breach and integrity of the system, and very often the revision of security aspects is concentrated in four basic stages:

- The **occurrence** of events,
- Event **detection**,
- Event **tracking**, and
- **Archiving** events.

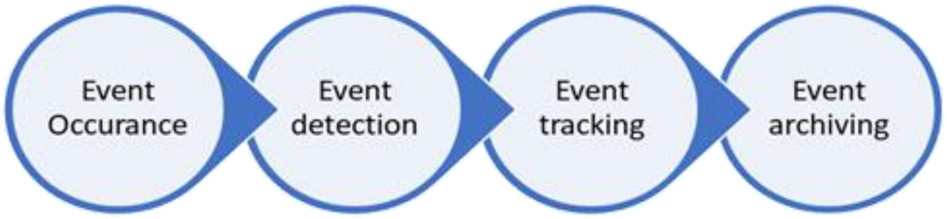

Figure 1. Audit of security incidents – flow diagram (author)

The **occurrence** of events often passes unnoticed in the system, since the challenger can be a part of the system (for example, a worker). However, it is very important that the system makes the event **detection**, or "to notice" that something is wrong, that is, that there is a latent (hidden) danger of changing, losing or misusing data or systems. The difference between occurrence and event detection, in time, can be significant, that is, relatively fair time from occurrence to detection of a particular event or incident. What does "relatively long time" mean? Such a statement is placed because very often this time frame is very small. That is, it is a very short time since the occurrence of the incident to the manifestation of the negative effects and the impact of that incident. In other words, depending on the severity and complexity of the incident, the negative effects can be manifested in a very short period of time, when the attacker is in direct contact with the system, or in a longer period of time when the attacker is waiting for a favourable moment to interact with the system and activate the unwanted an event (for example, an attacker is waiting for his working hours to access the system in order to avoid direct control). It is precisely because of this different approach that the detection of the phenomenon is very important, immediately after its emergence, if this can be achieved using an intrusion detection system.

Then the suspicious event is **monitored** to see its functioning (what is happening) in order to plan the way to eliminate the negative effects of an event or incident. Very often, this is the shortest phase because it should not be allowed the event to advance. The IDS contains certain predefined models of attacks that compares with the established and performs corrective measures already exist. However, it also happens that attacks or events are not recorded in the IDS, so it is necessary to react sensitively to the given incident. In general, it is avoided to allow further development of the incident, so the system reactions are mostly prompt and they occur immediately after identifying a negative phenomenon or intrusion and recorded in the relevant log files.

The system chooses an adequate response and neutralizes or prevents the further negative impact of event by various methods at its disposal and determines whether the incident is neutralized. If this is the case, the system **archives** an incident including archiving an event with the corresponding corrective measures, as well as adding an event to a list of records that will allow the system to prevent the re-occurrence of the same or a similar event.

In this regard, expert systems have the possibility of "*learning on the example*" and thus the possibilities of improving the technique of protection or detection of undesired changes in the system. These systems must also be designed to provide the possibility of preventive, online detection of intrusion into the system (or intrusions) in order to ensure timely protection of the system and data.

Dening proposes that such systems should be resident in the computerized accounting system, and they aim to monitor the behaviour of system users [3]. This should not be some *ad hoc* schemes that they provide a periodic check service, but the systems that constantly listen to communication and try to detect anomalies and irregularities. Dening has hypothesized that system exploitation also contains elements of abnormal use, so a security breach can be detected timely.

Examples of security breaches Deming states are:

- **Attempted decline** - a deliberate decline in the system that would allow the generation of more illegal credentials;
- **A salient decline** - represents a deliberate decline in order to make changes in the system, and the consequence of some lack of system security;
- **Penetration by a legitimate / registered user** - a common security activity where a registered user can deliberately or unintentionally endanger the integrity of the system and data, or may initiate a sequence of routines that are not in accordance with regular procedures;
- **"Leakage" of data by a legitimate / registered user** - an activity that also happens often, e.g. transmission of data by means of a transmission medium without prior authorization for such a move, printing on a network printer when it is contrary to the protection rules or "borrowing" its credentials to access the system to persons not authorized to access the data and system;

- **Virus** - also a very common phenomenon, which can manifest in several ways and in multiple manifestations;
- **"Phishing"** - the acquisition of a user's identity by hackers or other malicious persons who are not authorized to access data and system through interception and other methods.

Checking whether the system was in danger is very often tested using the so-called zero behaviour, or a specially created profile in the system for each user and for each object in the system. Then the current behaviour of the system is compared with the zero and determines whether the behaviour of the system is within the limits of normal or not. However, the question that arises is whether the artificial intelligence systems (that need to determine an intrusion) are sufficient considering the number of variables they handle or is still a human the type of detector that is the most reliable. The solution is, as always, a hybrid character, or a combination of these two modes.

3. Functioning and architecture of expert audit systems

Audit environment is a unique and very complex environment for decision making. There are many sources of errors and inconsistencies that are unique to the audit environment. Personal computers and other changes in technology have had an impact and will continue to affect the audit environment. In addition, the audit environment is focused on the process rather than on the results.

In the discussion of the audit environment complexity, Hansen and Mesir [4] argue that the audit problems are of a non-deterministic nature and as such have a large number of solutions. It is therefore very difficult (in some cases, and almost impossible) to examine all the solutions and to choose the best, but such problems are often solved by using heuristics to find a good or satisfactory solution that is not always the best and most optimal. Expert systems in this case have a key role precisely because of the involvement of heuristics in decision making, which will offer an alternative and feasible solution.

Hogart's study [5] showed that computer programs such as expert systems are very useful for improving the consistency of human thinking and reducing errors. Dillard and Macler [6] also noted that expert systems can contribute to consistency, thoroughness and credibility in the decision-making process on auditor's opinion. Expert systems are designed to provide continuous, online detection of downtime and system protection. Such systems are usually resident in computer systems and serve to monitor the behaviour of users of the given system [7].

The description of the architecture of expert systems for detection of network intrusions is mostly taken from the original document that started the development of these systems, which was developed by the group of scientists of the SRI International Computer Science Laboratory for the needs of the US Army. Although this architecture was made in 1995, it is still current and represents the essence of other systems that perform the same or similar functions.

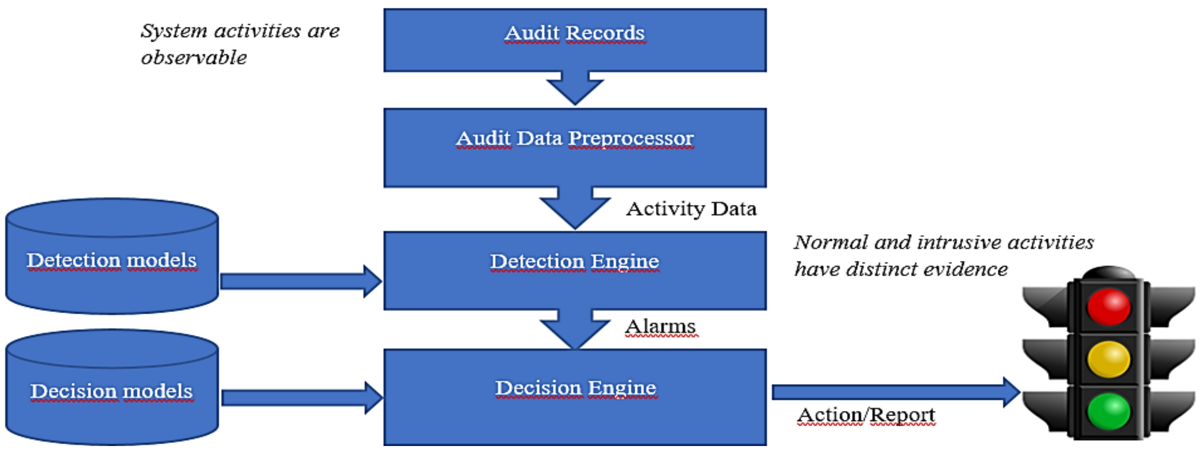

Figure 2. IDS Components (taken from [8])

The essence of NIDES is to connect several components that can interact with a remote call procedure (RPCs) or call the Library Calls. Anderson and others [9] define the following key components of NIDES:

- Persistent storage component,
- The program called *Agend*,
- *Agen* program,
- The program called *Arpool*,
- Component for statistical analysis,
- Analysis-based component analysis,
- The locking component (*Resolver*),
- Component for archiving results (*Archiver*),
- Batch analysis component
- User interface.

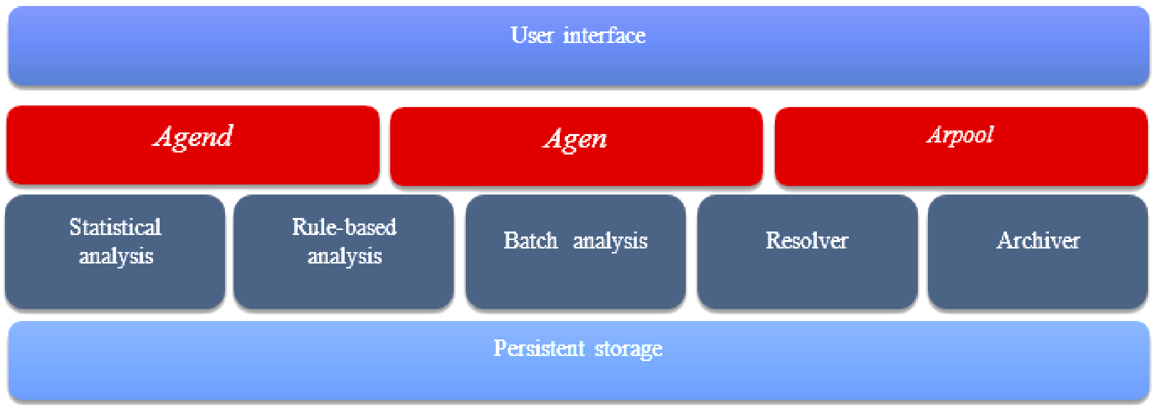

Figure 3. Four-layer NIDES architecture (adapted from [8])

The **Persistent storage component** includes a set of functions based on libraries that essentially perform the data management function for the needs of the NIDES process. The data contained in the permanent storage component are the archive data of the audit records, the archive of results, the user's statistical profile, and the analysis of the configuration information. This data set is created and stored for all instances of NIDES. NIDES instances use the concept of storing different versions of the same type of information. For example, each test that is made gets a unique name and represents a NIDES test instance that contains all the results, user profiles, and configurations analysis. Each test made has its own instance with which it is connected, and it is possible to re-use later, e.g. for multiple system tests. A special instance of NIDES is "Real Time" reserved for storing NIDES information in real time. This instance also contains all the elements - results, profiles, configurations, etc.

The **program named *Agend*** represents the background process (*daemon*) that is compulsory running on all targeted NIDES host systems. This program starts at every system boot-up for all hosts controlled by NIDES. *Agend* program works as the RPC server process and listens well-known ports regarding the request to start and stop the program for converting native auditing records (*Agen*) - that starts or stops through the user interface.

The ***Agen* program** is performed on each target host system that actively provides data for NIDES analysis. The work of this program is fairly easy to explain, although it is very complex in nature. The *Agen* program reads the original audit data, and then converts them into audit records in NIDES. After a successful conversion, it sends these records to the next program that processes them, which is the *Arpool* program. During the publication of the work (1995), NIDES functioned on the UNIX operating system that owned *Agen* programme too and supported three native audit data types: Sun OS BSM version 1, Sun OS C2, and standard UNIX accounting. The extension functionality has also been developed using the *Perl* script that has enabled users to develop their *Agen* scripts for other data sources. The task of this program is to process all available data types after the user prompts the user through the user interface and *Agend*.

The ***Arpool* program** processes the collected audit data from different target hosts and sends them to components for the analysis and detection of anomalies.

The **statistical analysis component** maintains historical statistical profiles for each user. If the system notices the activity of an individual that deviates from the established use, this component is triggering an alarm. In order for the component to function properly, it is necessary to regularly update historical profiles to help NIDES adaptively learn the behaviour of each user. The statistical analysis component can thus detect the incidents that are masked as legitimate users. On the other hand, statistical analysis can also detect attacks that use previously unknown vulnerabilities of the system, which cannot be otherwise detected. Statistical detection of anomalies can cause interesting and unusual events that can lead to discovery related to system security. Statistical analysis is flexible because the parameters and sense threshold can be easily changed from their default values, while specific crack detection measures (the behavioural aspects for which statistics are maintained) can be switched on or off. A series of papers has elaborated these statistical algorithms in detail. [10], [11]

A **rule-based analysis component** uses rules that characterize and describe known types of intrusions to enable the alarm to be triggered if the activity observed corresponds to any of the encoded rules. This type of analysis aims to uncover attempts to exploit known security vulnerabilities of supervised systems and intruders that show specific patterns of behaviour known to be essential or to violate the security policy of the system. Thus, any observed activity that corresponds to any of these predefined behaviours signifies. The rule database could be configured using the *rb_config* file, which supports the user configuration of the existing NIDES rules. The ability to customize the database of rules has been increased so that new rules can be defined and collected in an existing system, and existing rules can be turned on or off. Although the NIDES basic database is created for Sun UNIX operating systems, users should adjust the rule base for their environment and continually update the rules by adapting changes to the vulnerability of new system release and the newly discovered vulnerabilities of current releases.

The **resolver component** checks the alarms generated by the intangibles for analysis before registering them. Typically, one user action can generate a significant number of audit records. Therefore, an unusual action can lead to several tenths of hundreds of alarms reported by statistical analysis in a fast sequence. In order to avoid flooding the security administrator with redundant alarms, this component filters alarms to remove such redundancy. Alerts can be reported to NIDES consoles or emails lists, but some configuration filters can also be provided. For example, the administrator can shut out warning reporting for certain users if they know that they will do something unusual and that would otherwise generate a lot of false alerts. Although the filtered alerts have not been reported, they are have been recorded in the log.

The component for archiving the results controls and stores the audit records, the results of analyses and warnings. Archiving is possible through the user interface.

The **batch analysis component** allows the security administrator to experiment with new settings of statistical parameters or rules database settings before executing them as real-time operations in NIDES, and test results are archived for future comparison.

The **user interface** serves NIDES users to use all of the functionality. The user interface has a series of elements: drop-down menus, "*mark-and-click*" (enlarged *Point-and-Click*) selections and occasional text inputs. The interface also has a comprehensive multilayer context-sensitive help system, as well as system monitoring functionality that displays information about the supervised system, the status of the archive of audit data, an overview of the flow of the system every hour, and an overview of the generated alerts every hour.

Following is a brief explanation of the process itself, based on single event, while there is also possibility for batch event processing. The *Arpool* process retrieves data from multiple sources and combines them into a single stream of audit records by assigning a unique number of sequences to each record during this process. The NIDES generic format of the audit record can be easily adapted to new types of systems by writing simple routines for mapping audit data or using Perl scripts. Analysis components (statistical and rule-based analysis) receive audit data from the *Arpool* process that delivers audit records to all its users (those processes that have made the connection to obtain auditing data), and then discard them. Once the components have been analysed by individual audit records, these results are supplied to the locking components. The locking component analyses the results of both analytical components to determine whether a warning should be sent. If there is a need, this warning is sent to the user interface provided that the user has included at least one of the alert warning mechanisms (usually a popup notification window). Finally, all warnings issued are archived in the NIDES archive.

The basic problem of expert systems in the field of accounting data is the conditionality of their application and the domain character of expert systems. The use of the knowledge base is conditioned and based on a certain kind of symbolic logic that this knowledge can be presented. Thus, for example, logical programming is based on deduction, which also determines the goal of the program to make the problem applicable to deductive processes. The problem with the bases of knowledge of accounting and auditing information and data is not so pronounced when it comes to classical subsystems (auditing and accounting standards, etc.). But in the case of materiality or audit risk assessment, there is a significant problem due to their logical structures. In other words, such a form of knowledge may be more or less probable, and such knowledge is difficult to classify. It is questioned how the program will perform the classification of objects, regardless of which program classification rules will apply to the process of matching knowledge within the knowledge base.

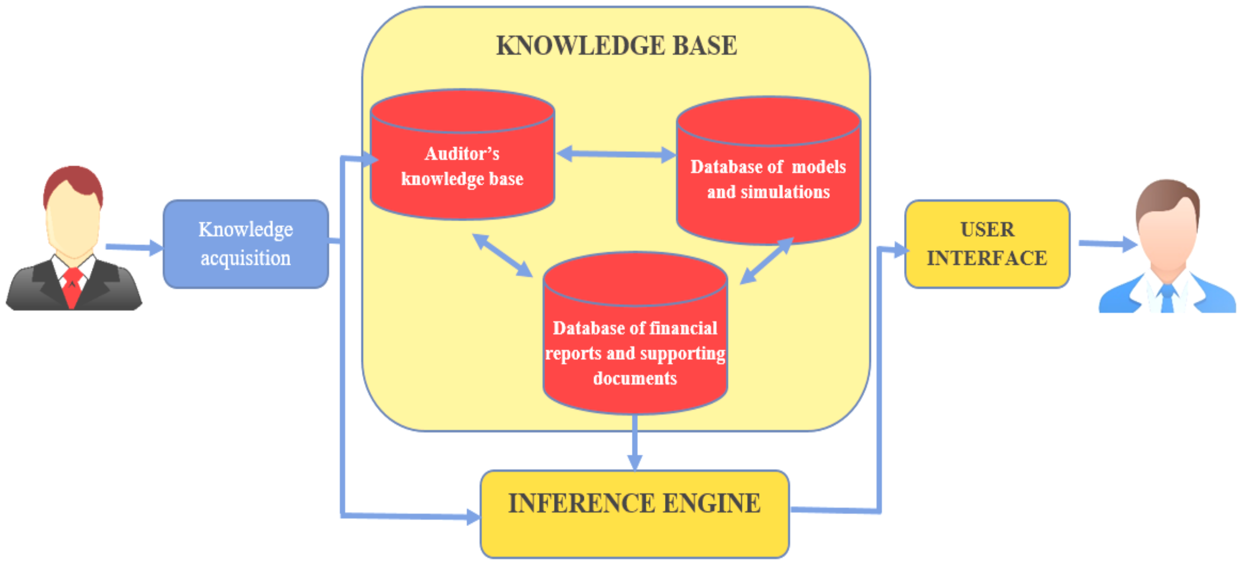

Figure 4. Generic model of expert systems' architecture for financial audit (author)

In the literature, three basic approaches to the development of this type of expert systems can be found:

- **Access to the Blackboard system approach [12-14],**
- **Frame-based approach [15],**
- **Object-oriented approach [16]**

The development of the system through **the "black board"** emphasizes the development of a set of independent domain specific modules called sources of knowledge, which interact with a common - shared data structure - "black board". In spite of the fact that it provides modularity, this approach does not say that a specific part of knowledge is being developed by other sources of knowledge - the "black board" does not "know" anything about the nature of the communication semantics. Just setting the hypothesis is a very limited form of communication when a comprehensive purpose is to mimic the processes of human thinking. Also, decisions on the further execution of the hypothesis are taken by an independent "planner", regardless of the knowledge of the hypothesis, which can reduce the coherence of the entire system [17].

A **frame-based approach** provides a more structured presentation of knowledge in the form of a framework. The box describes an object, consisting of slots containing the default values, pointers to other frames, rule sets, or procedures. Frames are related to ensuring inheritance and communication by transmitting messages. However, the modularity of knowledge presented in the frames can not be clearly defined, and the presentation lack flexibility [18]. Also, frame-based systems do not provide a way of defining fixed slots [19].

An **object-oriented approach** is an extension of a framework-based approach that allows the development of autonomous objects that communicate by transmitting messages to each other during troubleshooting. The mechanism for the transmission of messages in an object-oriented approach does not contain a provision on how the recipient must receive the message from the background and perspective of the object. Since human communication contains a context and intention other than content, only sending the content of a message becomes a very limited form of communication. In addition, the inclusion of new objects must be in line with the global scheme of the entire system, which can prove to be unfathomable in large and evolutionary systems.

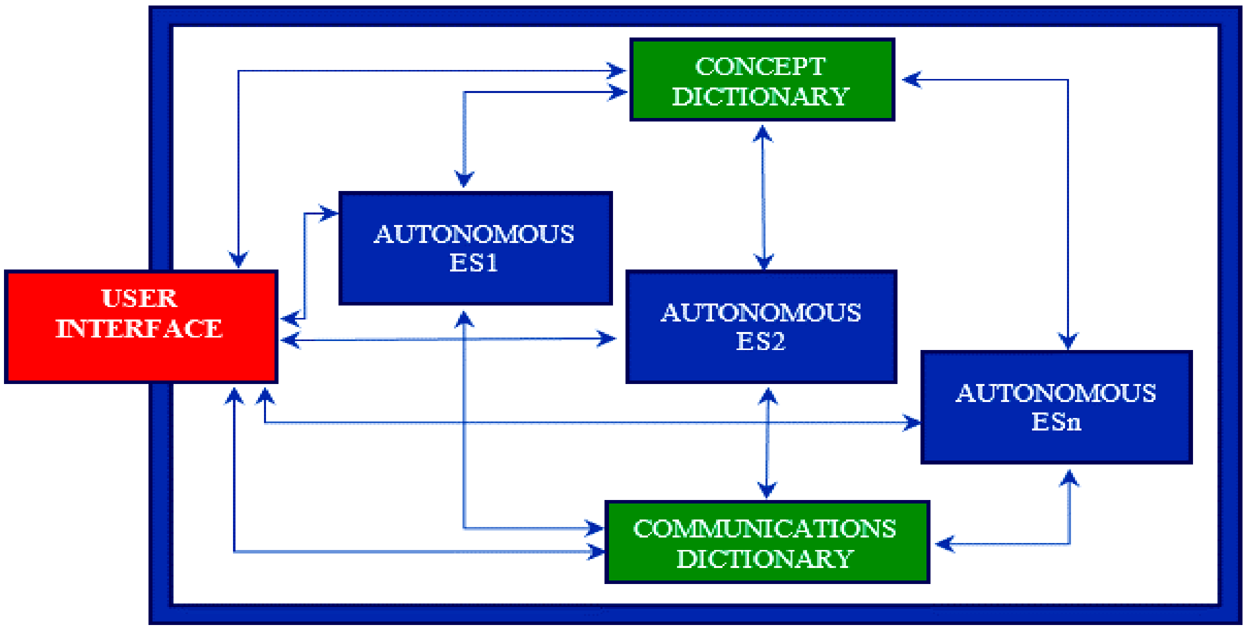

Figure 5. A diagrammatic scheme of the open architecture (adapted from [20])

In their work Kaula and Lander gave their vision of an alternative approach that would facilitate the development of large-scale expert systems to overcome the constraints of the aforementioned development approaches [20]. Their approach is based on the development of the system according to so-called ***Open-system approach*** which implies the development and existence of a system consisting of a large number of independently developed smaller subsystems that communicate and collaborate by forwarding messages (these messages can be knowledge, data, information) in the process of solving the problem [21], [22].

An open-based expert system architecture (OES) architecture is an expert system consisting of several independently developed smaller **autonomous expert subsystems (AESs)** that communicate during problem resolution. Each participant subsystem owns its own knowledge, i.e. assumptions, limitations, and rules, and each is responsible for maintaining its own internal consistency. No subsystem can directly control the knowledge of the other, making communication and negotiation essential for solving the problem. Also, modularity is ensured, since adding or omitting the subsystem minimizes the impact on the work environment. Also, communication between the subsystems is facilitated by the use of a communication dictionary that contains procedures for implementing communication protocols called „the message acts“.

Such system architecture is very suitable for the construction of an expert systems in the field of auditing and accounting. Open architecture provides scalability and a modular approach to construction that allows for caste or system customization to audit or accounting needs. Namely, such an open architecture is very suitable for large economic systems consisting of several profit centres that need to be audited or controlled. The shell system can be modified in this way, depending on the complexity and size of the business entity or institution.

Such an expert system architecture is suitable even for the construction of an expert systems that would function in public administration systems, where each separate expert subsystem could represent a specific ministry, agency or unit of local self-government.

4. Conclusion

The beginnings of artificial intelligence date from the ancient past. All the time, one wondered if machines (computers) could dress and think instead of man and make decisions? With the advent of computers in the middle of the XX century and the understanding of the possibilities they open, the question has changed in - will the machines replace the man and his intelligence? Today, even the most famous scientists such as Stephen Hawking warn of the dangers of accelerated development of the artificial intelligence system. But the general impression is that human imagination and knowledge do not have a decent rival in the world of computers for now because it is created by the same man.

However, one of the founders of the artificial intelligence of Minsk, in his works, claims that the machines are not aware of what they are doing, but they are not aware of human beings either. Expert systems, as part of the artificial intelligence family, were made to make certain decisions instead of man. As such, they could not do this without the influence of a man - an expert, who was supposed to bring his knowledge into a system in the form of an understandable computer, which based on such a "learning" could then "draw" conclusions and make decisions based on this knowledge.

Expert systems have, first of all, found their application, except in the military industry, from where they originated, in the domain of economics, more precisely accounting. Namely, almost at the same time as they were created, they were immediately applied to solve economic problems, especially in the domain of accounting and auditing. Manipulation of a large number of numerical data required the auditor a long time to find irregularities or abuses.

This paper explains the basic concepts of artificial intelligence and expert systems, the revision and revision of information systems and attempted to make parallels between them in a historical context. Unfortunately, the author was not able to access the most recent research in this field because the results of these surveys are charged, so he had to satisfy the sources available on the Internet. Therefore, it is very possible that the presented models of expert system architecture are not very difficult, but they certainly reflect the basic concept of organization and implementation of expert systems in the field of accounting and auditing.

It is certain that further development of this area is expected, since the field of control and audit is an important area for each organization or institution, along with the overall security of the information system within which the financial module is located. Therefore, this paper put emphasis on the presentation of expert systems that perform system monitoring and the prevention of intrusion into the system, as well as expert systems that help the auditor to find potential problems and abuses in a large amount of data. There are still many other issues open in this field, but certainly the use of neural networks and genetic algorithms is something that will continue to be studied, and it can be expected that these systems will inevitably be part of future plans and training programs in the newly emerging area, which is called science Data Science.

References

[1] Ramos M 2004 Section 404 Compliance in the Annual Report, *Journal of Accountancy* **10** 43-48
[2] Dunn C L, Cherrington J O and Hollander A S 2005 *Enterprise Information Systems: A Pattern-Based Approach*, 3. ed., McGraw-Hill Irwin, Boston, MA
[3] Denning D 1987 An Intrusion Detection Model, *IEEE Transactions on Software Engineering* **13**(2) 222-232
[4] Hansen J V and Messier W F 1982 Expert Systems for Decision Support in EDP Auditing, International Journal of Computer and Information Sciences **11** 357-379
[5] Hogarth R 1985 *Judgment and Choice*, Wiley, Chichester, USA
[6] Dillard J and Mutchler J 1987 *Expertise in Assessing Solvency Problems*, *Expert Systems*, 170-178
[7] O'Leary D E and Watkins P R 1989 *Review of Expert Systems*, in Auditing, Expert Systems Review, University of Southern California, USA
[8] Stolfo S 2017 Teaching presentation slides, Columbia University New York, Columbia Engeeniring The Fu Foundation School of Enginnering and Applied Science, Computer Science Department, https://www.cs.columbia.edu/~smb/classes/f06/l19.pdf
[9] Anderson D, Frivold T and Vlades A 1995 Next-generation Intrusion Detection Expert System (NIDES) - A Summary, SRI International, Computer Science Laboratory SRI-CSL-95-07
[10] Harold J S and Valdes A 1994 The NIDES statistical component description and justification, Annual report, SRI International, Menlo Park, CA
[11] Valdes A and Anderson D 1994 Statistical methods for computer usage anomaly detection using NIDES, In Conference on Rough Sets and Soft Computing
[12] Balzer R, Erman L D, London P E and Williams C 1980 Hearsay-III: a domain-independent framework for expert systems, Proceedings AAAI
[13] Erman L D, Hayes-Roth F, Lesser D R and Reddy V R 1980 The Hearsay-II speech-understanding systems: integrating knowledge to resolve uncertainty, *Computing Surveys* **12** 213-253
[14] Nii H P 1986 Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures, *The AI Magazine*
[15] Minsky M 1975 *A framework for representing knowledge*, in Winston P (Ed), The Psychology in Computer Vision, McGraw-Hill, New York, pp 211-217
[16] Yoon Y and Guimaraes T 1992 Developing knowledge- based systems: an object-oriented organizational approach, *Information Resources Management Journal* 15-32
[17] Kaula R and Ngwenyama O 1990 An approach to open intelligent information systems, *Information Systems: An International Journal* **15** 489-496
[18] Leung K S and Wong M H 1990 An expert-system shell using structured knowledge, *Computer* 38-47
[19] Giarratano J and Riley G 1989 *Expert Systems: Principles and Programming*, PWS-Kent Publishing Co, Boston, MA
[20] Kaula R and Lander L C 1995 A module-based conceptual framework for largescale expert systems, *Industrial Management & Data Systems* **95**(2) 15-23
[21] Hewitt C and de Jong P 1984 *Open systems*, in Brodie M L, Mylopoulos J and Schmidt J W (Eds), On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages, Springer-Verlag, Berlin, pp 147-164
[22] Kaula R 1990 *An open intelligent information systems architecture*, PhD Thesis, State University of New York, Binghamton