SILESIAN UNIVERSITY OF TECHNOLOGYPUBLISHING HOUSE

SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY 2024
ORGANIZATION AND MANAGEMENT SERIESNO. 212

# ANALYSIS OF THE OPTIMAL ORDER OF DIAGNOSTIC ACTIVITIES USING AN IT TOOL – A CASE STUDY

Adam GUMIŃSKI[1*], Wojciech ZOLEŃSKI[2], Łukasz GUMIŃSKI[3]

[1] Silesian University of Technology; adam.guminski@polsl.pl, ORCID: 0000-0001-8250-7101
[2] Silesian University of Technology; wojciech.zolenski@polsl.pl, ORCID: 0000-0001-7336-2318
[3] Silesian University of Technology; lukasz.guminski629@gmail.com, ORCID: 0009-0004-3953-5034
* Correspondence author

**Purpose:** The article deals with the problem of supporting a decision-making process in manufacturing enterprises. It concerns the possibilities of software algorithm determining the optimal path of knowledge acquisition in the form of an expert system. As part of the research, the conditions for developing the inference module and the knowledge base were determined. In order to assess the usefulness of the analysed expert system, its validation and simulation analysis of determining recommendations regarding decisions made by system's user were carried out.
**Design/methodology/approach**: The research include the following tasks: literature study in the field of expert systems, analysis of existing IT solutions supporting decision-making processes in manufacturing enterprises, analysis of applied expert systems and conditions for their implementation, determination of design assumptions in terms of functional and non-functional requirements, preparation of expert system architecture, development of a knowledge base in the expert system, development, programming and verification of the expert system's inference module, formulation of final conclusions and indication of directions for further development of the expert system.
**Findings:** The study resulted in the development of the IT tool supporting a decision-making process in a production process. The developed expert system is intended to provide the expert with advice on the knowledge acquisition path that he should take in order to minimize the cost of inference. In order to perform this task, the system performs simulations and conducts the inference process on them, checking which path, with the parameters provided by the expert, is the best. The core of the application is the inference module of the system. It is used to perform simulations and derive the canonical form of the final hypothesis, and to determine the nesting levels of individual statements.
**Research limitations/implications:** At the current stage of development, the application's capabilities developed in the research, are limited. This is due to the fact that the database of statements that are interrogative conditions can only consist of a few dozen statements and the number of depths cannot exceed 10 levels, because the computational load becomes so serious that conducting a simulation would take too long for the expertise to be useful. In the current state, the increase in the load on the machine on which the inference process is run grows exponentially with the addition of subsequent statements that are interrogative conditions.

Further development of the application depends on finding a metaheuristic that will provide a solution that meets the requirements in a sufficiently short time. The solution to this problem might be achieved with a genetic algorithm.
**Originality/value:** The value of the research is the identification of elements determining a decision-making process in a production enterprise. The conducted research resulted in developing and implementing the expert system, the task of which is to provide support in the decision-making process based on the transferred knowledge. The application module is responsible for interpreting the knowledge entered into the system and transferring the processed knowledge in the form of specific advice.
**Keywords:** IT solution, expert module, optimization of hypothesis verification cost.
**Category of the paper:** research paper, case study.

## 1. Introduction

The changes taking place in the global economy mean that companies are forced to adapt more and more to the internal and external conditions of their operations. The key to success in many industries is the right decisions, made at the right time, within the scope of specific activities. This applies in particular to manufacturing companies, due to the high complexity of the technological processes implemented, which require advanced technology and technical infrastructure. The decision-making process requires support from IT tools, which are largely based on expert knowledge. Avoiding incorrect decisions and minimizing inefficient activities is crucial for the production environment. Due to the specificity of its activity, each manufacturing company has a strictly defined production profile, which requires a specific technical infrastructure, technical and technological solutions and competent staff. Taking into account the criterion of the stability of the functioning of a manufacturing company, special attention should be paid to ensuring the continuity of the production process in relation to the existing technical and organizational capabilities. IT support for production management processes requires strictly defined data processing. Taking into account the functional scope of the serviced domain areas, the following can be distinguished (Klonowski, 2004):

- partial systems that are used within a single field and include selected utility functions;
- single-domain systems, which usually include all the functional functions required within a given field. The use of such a solution usually does not require computerization of other departments or modification of the structure of information and decision flows in a production enterprise. There is a situation in which input data to single-domain systems is entered directly by the operators of these systems;
- multi-domain systems, which offer integrated functions in at least two subject areas. These systems usually have different domain scopes and degrees of integration. However, to achieve full system integrity, integration at the database level is necessary. Using system integration significantly reduces the workload associated with entering data and ensuring its currency and consistency.

For determining an effective decision in the field of production system diagnostics is the determination of the optimal path of taking actions, which allows for reducing operating costs and the time needed to make the final decision. Considering the criterion of maintaining the continuity of the production process, the process of technical support of the production system is becoming increasingly important. The basic tasks of this process include (Pająk, 2006):

- achieving failure-free operation of production equipment and machines while maintaining their characteristics, which enables ensuring the needed quality of products,
- minimizing losses resulting from potential failures of machines and devices and costs related to the repair of these machines and devices,
- maintaining safe operating conditions for machines and devices. The production system service is focused on activities aimed at minimizing potential damage to technical equipment.

The essence of technical service and diagnostics of the production system becomes the more important the more complex the production system is, both in terms of the number of machines and devices operated, as well as the scope and scale of the technological processes carried out. Technical diagnostics includes the following processes and tasks:

- monitoring, i.e. observation of quantitative and qualitative changes in values that characterize the course of operation of machines and technological devices without affecting the device,
- supervision, i.e. processing and transmitting information about the technical condition of machines and technological devices and influencing these devices in order to obtain the desired condition,
- diagnosis, i.e. detection of any abnormalities or an increased probability of their occurrence.

The expert system can be used in manufacturing enterprises, both those where serial production takes place (e.g. car production) and those where mass production takes place (e.g. hard coal production). The problem to be solved is of a praxeological nature. It concerns the possibilities of software algorithm determining the optimal path of knowledge acquisition in the form of an expert system. As part of the research, the conditions for developing the inference module and the knowledge base were determined. In order to assess the usefulness of the analysed expert system, its validation and simulation analysis of determining recommendations regarding decisions made by system's user were carried out. As part of the research, the following tasks were carried out:

- literature study in the field of expert systems,
- analysis of existing IT solutions supporting decision-making processes in manufacturing enterprises,
- analysis of applied expert systems and conditions for their implementation,

- determination of design assumptions in terms of functional and non-functional requirements,
- preparation of expert system architecture,
- development of a knowledge base in the expert system,
- development, programming and verification of the expert system's inference module,
- formulation of final conclusions and indication of directions for further development of the expert system.

## 2. The scope of application of expert systems as a theoretical background to the research

E. Feigenbaum noticed that the key factor that determines the level of quality of expert systems is knowledge that is recorded in the form of a knowledge base. The inference schemes contained in the expert system are of much less importance (Shortliffe, Buchanan, 1984). Therefore, the creator of the system is not forced to precisely determine how to solve the problem, but only to clearly formulate what the problem is and to gather knowledge that characterizes the problem to be solved in a declarative form. In contrast to procedural programs (based on data structures and computational algorithms), expert systems require expert knowledge contained in the knowledge base and an inference module.

An expert system can be defined as a computer program designed to solve specialized problems (Mulawka , 1997). An expert system can also be defined as a knowledge-based system in which knowledge and reasoning procedures are modelled after experts. In turn, the knowledge base itself is separated from the reasoning system and other modules of the expert system (Dohn et al., 2013). Researchers emphasize that it is necessary for the knowledge base (the part of the program containing expert knowledge) to be separated from the reasoning module (the part of the program solving the problem) in an expert system (Nieberliński, 2000; Gregory et al., 2016). Generally, it can be stated that expert systems are those computer programs that support solving problems requiring domain knowledge based on the experience of experts. Moving on to the characteristics of expert systems, their typical properties can be indicated (Zoleński, 2014):

- in expert systems it can be noticed that in order to solve the problem it is necessary to separate the knowledge base from the inference module and other system modules,
- advanced research is being carried out on automatic knowledge acquisition (e.g. inductive reasoning, case-based reasoning), but the main source of knowledge is the knowledge and experience of industry experts,
- the operation of the expert system is based on the use of logical models in which rules are applied in the form of conditional statements,

- the inference process there may be a dialogue with the user, who can analyze the provided partial results. In turn, the expert system may require supplementing the data in order to carry out further processing,
- knowledge is usually recorded in an explicit form, and several universal logical laws are applied in the inference module (Zieliński, 2000).

## 2.1. The structural and functional scope of the expert systems

The issue of expert systems, whose primary goal is to make an optimal decision based on codified knowledge contained in the knowledge base, has been studied since the 1970s. Analyzing this entire period, one can find numerous implementations of advanced and multi-module knowledge processing systems (e.g. CLIPS). It should be emphasized that the key problem of expert systems is to create a coherent knowledge base, which remains the most difficult and risky (from the point of view of return on invested financial resources) stage of developing an expert system.

To minimize the risk of incorrect knowledge input into the knowledge base, validation of inputs and consistency with other rules in the knowledge base has become increasingly important. The basic assumption when creating a knowledge base is to introduce subsequent rules, implemented by domain experts with unique knowledge in a specific field of the system. However, even such an assumption may be insufficient, which means that the introduced rules will be contradictory. Hence the need for validation to eliminate such a situation.

An important application of expert systems were implementations in the manufacturing industry. The main reason for their relatively high popularity was the need to consolidate expert knowledge, which was being lost due to natural reasons, such as the change of specialist staff from experienced to unfamiliar with the field. This resulted in large losses in the company, which could not be avoided without effective knowledge transfer. Such transfer can be ensured by expert systems. The role of the expert in consolidating knowledge focuses on introducing knowledge in the appropriate representation, which additionally requires in-depth knowledge of the IT system, which is rarely encountered. A much more frequent case is the transfer of knowledge from an expert to a knowledge engineer, who not only has in-depth knowledge of the IT system, but also has knowledge of concepts from the field of knowledge. He translates the expert's formulations into the expert system representation. After the knowledge base has been formed, it then becomes possible for new people to use the system. Additionally, it is possible to automate the acquisition of expertise through automatic measurements, which, after verification of correctness, can be used for inference by the inference module in the expert system.

The issue of expert systems, whose primary goal is to make an optimal decision based on codified knowledge contained in the knowledge base, has been studied since the 1970s. Analyzing this entire period, one can find numerous implementations of advanced and multi-module knowledge processing systems (e.g. CLIPS). It should be emphasized that the key

problem of expert systems is to create a coherent knowledge base, which remains the most difficult and risky (from the point of view of return on invested financial resources) stage of developing an expert system.

To minimize the risk of introducing incorrect knowledge into the knowledge base, validation of inputs and consistency with other rules in the knowledge base has become increasingly important. The basic assumption when creating a knowledge base is to introduce subsequent rules, implemented by domain experts who have unique knowledge in a specific field of the system. However, even such an assumption may be insufficient, which means that the introduced rules will be contradictory. Hence the need for validation to eliminate such a situation.

There are numerous advantages of expert systems. Separating the inference module and the knowledge base enables cheaper and easier creation of the system itself, as well as its subsequent modification or operation. Expert knowledge occurs in an explicit form, written in a language close to natural, which results in its ordering and structuring. The knowledge base can be updated or expanded without violating the integrity of the system (Nieberliński, 2000). Usually, the knowledge contained in the bases is in the form of multi-level nested rules, which can be extended with subsequent exceptions or new rules, which allows for the development of knowledge bases in an incremental manner, depending on the existing needs regarding the scope and detail of domain knowledge. It is also possible to create skeleton systems (shells), which are inference modules without knowledge bases (Nieberliński, 2000; Evans, 2004). Creating knowledge bases in an expert system is much easier, and thus can be implemented by knowledge engineers working in the company, without the interference of external companies. Expert knowledge is of great importance for the functioning of the company, which is difficult to obtain, and without its proper storage it can be irretrievably lost. Expert systems have the advantage of enabling its collection, as well as its verification, correction, development, appropriate ordering and categorization. Knowledge stored in databases can be used multiple times and transferred to different recipients. Multi-stage verification of knowledge using an expert system means that codified knowledge can be at a higher level than the knowledge of individual experts. Reducing the values of variables to binary form allows to avoid combinatorial explosion (Kowalski, 1989). For example, instead of recording the exact temperature of a device, it may be enough to have information whether its temperature is within a given range. Logical models used in knowledge databases - binary information may be sufficient for the inference module. In inference modules, instead of two-valued logic, approximate inference can be used. A statement may then have an approximate truth or approximate certainty - such a record often allows for a better representation of the hidden knowledge of the expert, which in turn allows for a better description of the problem model and available data. Approximate inference combines some of the advantages of logical models and continuous models. The disadvantage of approximate inference is the greater complexity of knowledge and the inference system. Quantitative variables can be interpreted qualitatively, taking into account the most important features of the data, their classification and selection (Gołuchowski, 2007). An expert system can draw conclusions based on knowledge with a low level of structure, for which there are no analytical mathematical models. In such situations, the space of solutions is finite, which allows for exploring the entire space. On the other hand, when the size of the space is large, blind and heuristic search algorithms should be used (Cytowski, 1991). Expert systems can also use knowledge in a declarative form, which is easier to describe and formalize. However, such an approach requires experience, because if the specific features of the applied search methods are not taken into account, the inference module may hang even in simple and correctly formulated tasks (McCabe, Clark, 1998).

### 2.2. Conditions of the implementation of expert systems

The most important issue in the design and implementation of expert systems is the process of acquiring knowledge from domain experts. Limited knowledge of these systems results in a deficit of specialists/experts in the field of knowledge engineering who have the appropriate experience in acquiring and verifying the tacit knowledge of experts (Cholewa, Pedrycz, 1987). The process of implementing an IT system in an enterprise is always risky. In the literature, one can find the six-person-month rule, i.e. "if the time and human involvement at this level does not bring satisfactory results, further work should be stopped, accepting the loss of the previously incurred expenses, because the losses may significantly increase" (Zoleński, 2008). A similar situation can be observed in the implementation of expert systems. It is extremely important to clearly define the appropriate application of the expert system. A cost-effective implementation takes place if the problem to which the expert system is to be applied meets clearly defined requirements. Most problems occurring in practice do not meet these requirements. The progress of technology, reduction of the costs of implementing computer systems and increasing awareness on both sides of the implementation should lead to improved efficiency of expert system implementations.

The fundamental problem in implementing expert systems is the excessive expectations of future users in terms of the scale and scope of expert system functionality, which makes it difficult to reach a compromise that satisfies both parties to an IT project. Unfortunately, there is a widespread misconception that large expert systems comprehensively solve a wide range of problems with a high level of difficulty. Such an approach is associated with high financial outlays, which the majority of companies are not convinced of. The calculated level of risk for both parties is often unacceptable. In connection with the above, it is profitable to use expert microsystems for supporting a certain class of tasks, the knowledge bases of which are created ad hoc by users. Wider use of expert microsystems allows for much higher efficiency of implemented implementations with a significant reduction in financial outlays and risk. The use of microsystems builds the experience gained by employees in creating, implementing and using these systems, which can be used in the implementation of large expert systems. An additional benefit resulting from the implementation of microsystems are the knowledge bases created, which can be used in the future as components of an integrated modular system. The current benefits of using microsystems are usually not large, but an important effect is the experience gained by users in the field of knowledge engineering. Participation in the preparation and validation of the knowledge base, thorough knowledge of substantive issues and mastering the principles of formalizing, organizing and documenting knowledge, often has a much greater value than the current benefits resulting from the use of the microsystem. The most important features of an expert microsystem include (Dohn et al., 2013):

1. The knowledge base, created according to the specific needs of the microsystem users, is usually relatively small.
2. The knowledge representation is simple; it includes from a dozen to several dozen rules contained in the knowledge base.
3. The application of the microsystem is adapted to the current needs of users.
4. The expert microsystem has a structure in which the knowledge base and the inference module have been separated. The inference module usually comes from the skeleton expert system.
5. The expert microsystem is a link in a longer chain of data processing.

Expert microsystems, apart from supporting expertise, can also perform other functions, such as (Dohn et al., 2013):

- management of simulation and computational processes,
- automation of repetitive activities such as organizing information or automatically generating reports,
- implementation of logical operations,
- training users in the area of knowledge engineering,
- experimenting with inference modules.

The basic advantages of implementing expert microsystems include (Dohn et al., 2013):

- significant reduction of the risk associated with the implementation of the microsystem, financial outlays are fully acceptable,
- exploitation of microsystems enables gaining experience in the field of knowledge engineering,
- exploitation of microsystems defines further possibilities in the areas of potential applications for larger expert systems,
- archived knowledge bases can be used to create larger expert systems,
- employee potential can be better used. Implementation of microsystems for performing current tasks increases employee engagement, strengthens the bond with the enterprise and improves work efficiency.

## 3. The description of the research method - the expert system supporting a decision-making process in the production environment

The developed expert system is intended to provide the expert with advice on the knowledge acquisition path that he should take in order to minimize the cost of inference. In order to perform this task, the system performs simulations and conducts the inference process on them, checking which path, with the parameters provided by the expert, is the best. The core of the application is the inference module of the system. It is used to perform simulations and derive the canonical form of the final hypothesis, and to determine the nesting levels of individual statements. All these operations are intended to provide the expert with advice on the optimal or suboptimal path, the difficulty of obtaining the final hypothesis (in relation to the degree of nesting) and the statement (expert opinion) that should be asked about first.

### 3.1. The operation of request module

Due to the different costs of determining the logical values of individual statements, which are query conditions, it may turn out that the cost of asking about several statements (expert opinions) may be lower than the cost of asking about the value of one statement. The inference module therefore takes into account the costs of performing individual expert opinions (understood here as determining the value of the statement that is the condition of the rule) and is guided by the criterion of the lowest cost of obtaining the final hypothesis. For the inference module, this cost is an abstract entity, it does not have to mean a cost in the literal sense (money). The cost may consist of many different results:

- time of acquiring knowledge,
- impact on other elements of the system, e.g. acquiring knowledge may involve stopping production,
- financial cost of acquiring knowledge,
- difficulty of acquiring knowledge - in a production environment this may mean having to pump water out of a tank, turning off heaters in a room, etc.

Based on the cost, we can say that the system is therefore universal, because regardless of the industry, the definition of cost may be different. Therefore, the expert's task is also to determine such a cost function that will well reflect the field for which the knowledge base is implemented.

### 3.2. Use case diagram

The use case diagram includes system users and the functionalities that have been prepared for them. Each user has permissions to perform the actions indicated in the diagram (Fowler, 2005; Martin, 2014). The expert has the permissions of a standard user, as does the administrator. Only a guest can register - all the rest are already logged into the system (Authentication verified them as users with specific roles). The expert can also perform actions allowed for a standard user, as the expert role always goes hand in hand with the user role. Each user, except the administrator, can be blocked and deprived of access to the system and its functionalities. Similarly, unblocking can only be performed by the admin, not by changing the password or similar operation.

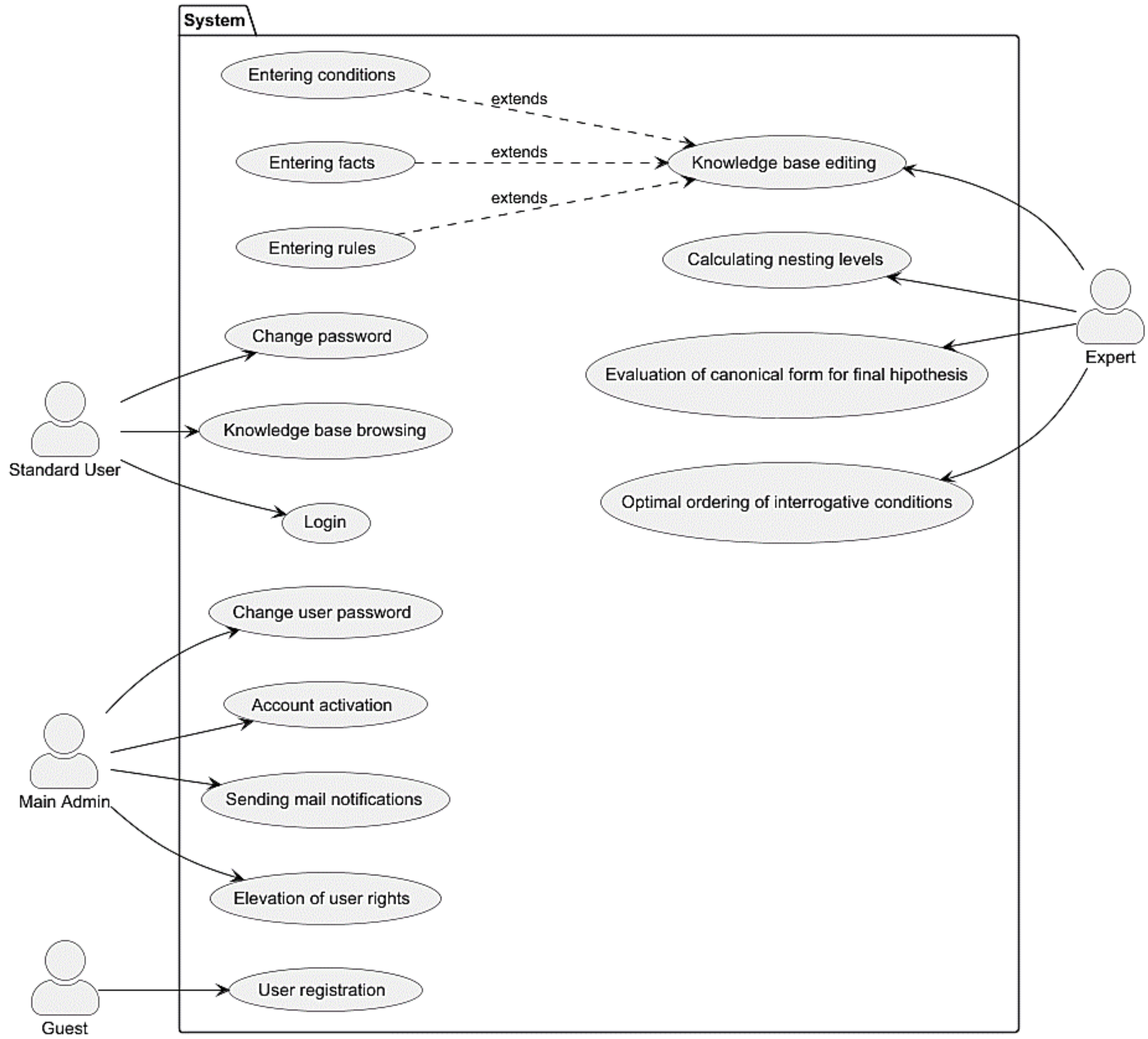


**Figure 1.** Use case diagram.

Source: own elaboration.

The expert system, in its inference part, performs the following 5 main operations:

- generating random permutations of query conditions along with their logical values,
- assigning nesting levels to statements,
- determining the statement about which knowledge should be acquired first in order to minimize the cost of inference,
- determining the optimal path for determining the value of statements by the expert,
- deriving the canonical form of the final hypothesis.

Determining the optimal path for determining the value of statements and the canonical form of the final hypothesis allows the system user to make a recommendation in order to achieve the lowest inference cost.

### 3.3. Generating a set of permutations

The basic algorithm of the reasoning system is the generation and determination of the optimal path of knowledge acquisition. This method consists of two stages:

- generation of random permutations of query condition sequences and, in the generated order, assigning logical values to statements until the final hypothesis is determined (the logical value will have a specific value: True or False),
- determination of the optimal path for determining the logical value of the final hypothesis based on the expected values of the expert opinions (according to the criterion of the least cost).

The size of the generated set of random permutations depends on the number of query conditions. The number of cases to be generated is determined by the formula:

$$n = 10^{a+1} \quad (1)$$

where:

n – the number of cases generated,

a – the number of statements that are interrogative conditions.

For a = 4 the number of generated cases will be equal to n = 100,000. This number of permutation cases allows for the repetitive acquisition of the same processing results of the inference module.

The inference module uses two-valued logic, where:

- 1 - stands for truth,
- -1 - stands for false.

We assume that 0 means no knowledge. Using such a representation greatly simplifies the evaluation of logical expressions at a later stage, especially since the rules in the knowledge base are Horn clauses. Therefore, any evaluation of a logical expression can be simplified.

For example:

$$A \land B \Rightarrow C \tag{2}$$

The logical value of statement C can be obtained as follows:

$$(C) = min(W\,(A),\ W\,(B)) \tag{3}$$

where W(x) - the numerical value of the statement with index i, lack of knowledge and no attempt to acquire it is marked as 0 - lack of knowledge.

### 3.4. Cost of acquiring knowledge - final hypothesis

The cost of determining the logical value of the final hypothesis is the sum of the costs of acquiring knowledge about the individual component statements (logical values of these statements). It is usually not the sum of the costs of acquiring knowledge about all statements, because examining just some of them can guarantee obtaining the logical value of the final hypothesis. For the permutation of statements - interrogative conditions: A, B, C, D, the cost of acquiring knowledge - the logical value of the final hypothesis will be as follows:

$$K = \sum_{i=1}^{a} \mathrm{K(i)} \cdot W(i) \tag{4}$$

where:
K – total cost,
a – number of statements that are interrogative conditions.
K(x) – the cost of acquiring the logical value (knowledge) of the statement with index i |W(x)| - modulo the numerical value of the statement with index i, lack of knowledge and no attempt to acquire it is marked as 0 - lack of knowledge.

For the knowledge base containing the above statements, the formula will look as follows:

$$K = K(A) * |W(A)| + K(B) * |W(B)| + K(C) * |W(C)| + K(D) * |W(D)| \tag{5}$$

## 4. The presentation of the results of an example of reasoning using the developed expert system

Carrying out the inference process requires first preparing input data in the form of rules and conditions (Fig. 2).

System configuration EXPERT_SYSTEM_USER

Statements Rules and conditions Estimation

Rules

| Label | Conclusion | Description |
|---|---|---|
| R1 | K | symptom a occurs |
| R2 | K | symptom a is absent and symptom is present |
| R3 | K | symptoms c and d occur |
| R4 | W | symptoms d and k occur |

Conditions

| Rule | if positive | Statement |
|---|---|---|
| R1 | true | A |
| R1 | false | B |
| R2 | false | A |
| R2 | true | C |
| R3 | true | C |
| R3 | true | D |
| R4 | true | D |
| R4 | true | K |

**Figure 2.** Rules and conditions introduced for simulation purposes.

Source: own elaboration.

The statement table shows the entered statements along with the levels that were calculated based on the given rules and conditions (Fig. 3).

System configuration EXPERT_SYSTEM_USER

Statements Rules and conditions Estimation

| Label | Interrogative condition | Description | Probability | Cost | Level |
|---|---|---|---|---|---|
| A | Yes | Device damaged | 0.2 | 200 | 0 |
| B | Yes | Device damaged | 0.4 | 300 | 0 |
| C | Yes | Device damaged | 0.3 | 100 | 0 |
| D | Yes | Device damaged | 0.1 | 400 | 0 |
| K | No | Component faulty | | | 1 |
| W | No | Component faulty | | | 2 |

**Figure 3.** Statements entered with nesting levels calculated based on rules and conditions.

Source: own elaboration.

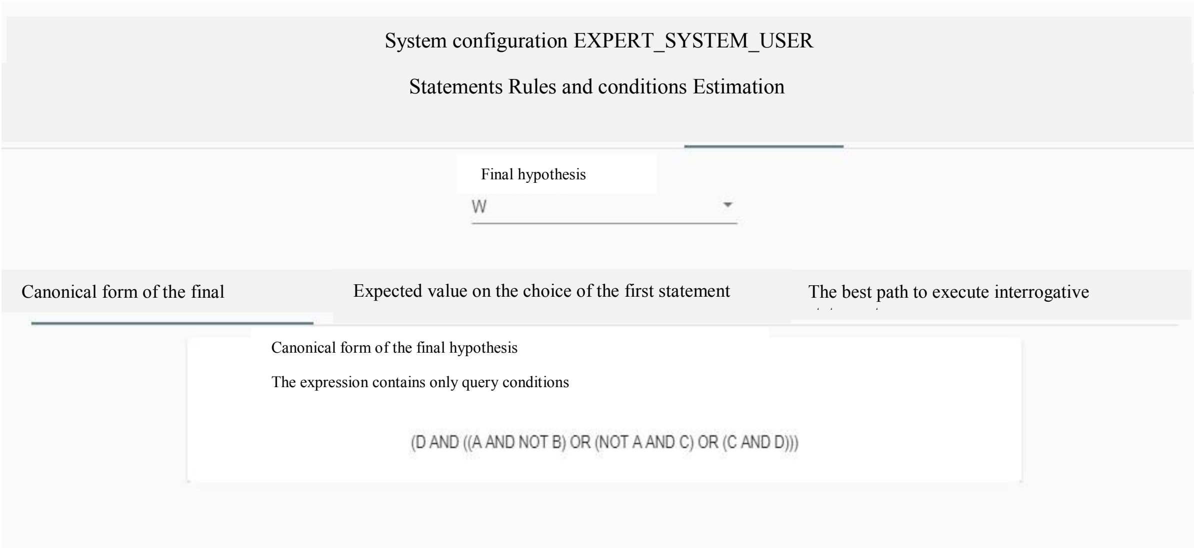


**Figure 4.** The entered statements along with the levels that were calculated based on the given rules and conditions.

Source: own elaboration.

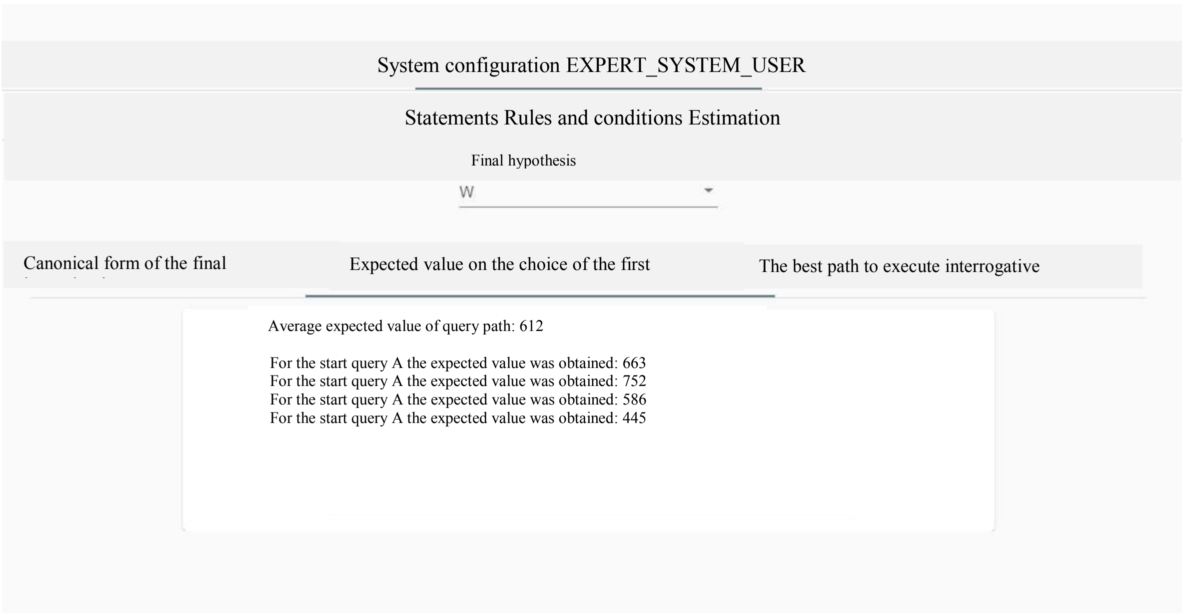


**Figure 5.** Presentation of the expected value of the final hypothesis evaluation.

Source: own elaboration.

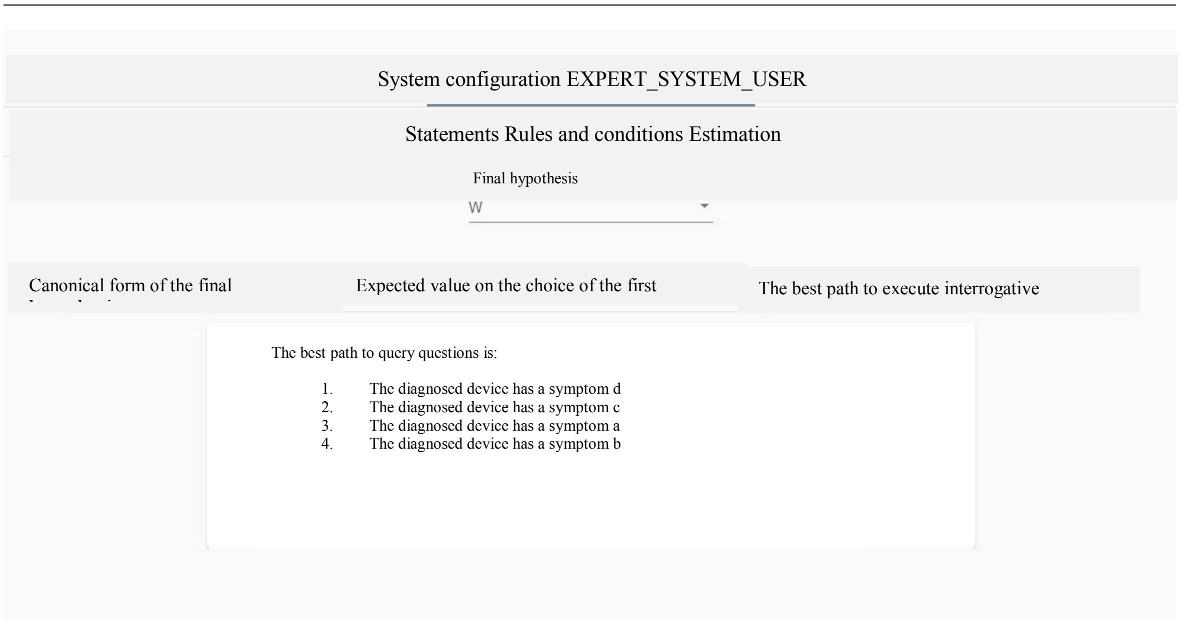


**Figure 6.** Presentation of the optimal path to determine the value of statements.

Source: own elaboration.

The analysis performed allowed us to establish the canonical form of the final hypothesis (Fig. 4). The statement table shows the entered statements along with the levels that were calculated based on the given rules and conditions.

In addition, the evaluation value of the final hypothesis was established (Fig. 5) and the optimal path to determine the value of statements – 427 (Fig. 6)**.**

## 5. Conclusions

To sum up the analysis and considerations undertaken in the research, the following conclusions can be formulated:

1. The dynamic development of the demand for knowledge determines research in the field of creating IT tools supporting knowledge management processes, including the creation of domain expert systems.
2. Improving decision-making processes requires IT support. In connection with this, we can observe a wide range of IT solutions that, in a more or less comprehensive way, enable automation and effectiveness of decisions and actions in business processes. Expert systems contain modules that constitute algorithmic solutions to problems that require mass data processing.

3. As part of the conducted research, an expert system was developed and implemented, the task of which is to provide support in the decision-making process based on the transferred knowledge. The application module is responsible for interpreting the knowledge entered into the system and transferring the processed knowledge in the form of specific advice.
4. As part of this work, a computer application was developed and verified that implements the following main functionalities:
   - creating new knowledge bases,
   - introducing statements and rules to knowledge bases,
   - conducting inference on knowledge bases,
   - evaluating the canonical form of the final hypothesis,
   - determining the optimal order of performing actions,
   - determining the optimal first interrogative condition,
   - evaluating the level of nesting of statements based on the rules and conditions introduced to the knowledge base.
5. Finalizing the application required the following tasks:
   - establishing design assumptions (requirements analysis, application architecture development),
   - establishing the database structure,
   - developing a use case diagram,
   - developing an ERD diagram,
   - developing a class diagram,
   - developing a knowledge representation in the database,
   - validating the knowledge base,
   - developing the expert system's reasoning module (description of the algorithm's operation along with its block diagram),
   - verifying the algorithm's operation.

## 6. Discussion on further development of the application

At the current stage of development, the application's capabilities are limited. This is due to the fact that the database of statements that are interrogative conditions can only consist of a few dozen statements and the number of depths cannot exceed 10 levels, because the computational load becomes so serious that conducting a simulation would take too long for the expertise to be useful. In the current state, the increase in the load on the machine on which the inference process is run grows exponentially with the addition of subsequent statements that are interrogative conditions.

Further development of the application depends on finding a metaheuristic that will provide a solution that meets the requirements in a sufficiently short time. The solution to this problem may be the use of a genetic algorithm (Kwaśnicka, 1998). A limited number of generated solutions, but on the other hand, also the diversity of cases provided by mutations and crossovers, will allow the user to return valuable advice on the decisions made, despite the large number of statements with the query condition flag. Another improvement of the system may be recursive validation of the existence of a loop, which can prevent the system from hanging up on the server side (to avoid a situation in which a statement in the rule condition will be a conclusion at a higher level of the same path, which would lead to the creation of an infinite loop).

## References

1. Cholewa, W., Pedrycz, W. (1987). *Systemy doradcze*. Wydawnictwo Politechniki Śląskiej.
2. Cytowski, J., Bolc, L. (1991). *Metody przeszukiwania heurystycznego*. PWN.
3. Dohn, K., Gumiński, A., Matusek, M., Zoleński, W. (2013). *Model wspomagania zarządzania w zakresie zarządzania wiedzą w polskich przedsiębiorstwach budowy maszyn*. Difin.
4. Evans, E. (2004). *Domain-Driven Design*. Helion.
5. Fowler, M. (2005). *UML w kropelce*. LTP Media Software.
6. Gołuchowski, J. (2007). *Technologie informatyczne w zarządzaniu wiedzą w organizacji*. Wydawnictwo Akademii Ekonomicznej.
7. Gregory, G., Bauer, C., King, G. (2016). *Java Persistence. Programowanie aplikacji bazodanowych w Hibernate*. Helion.
8. Klonowski, Z. (2004). *Systemy informatyczne zarządzania przedsiębiorstwem*. Oficyna Wydawnicza Politechniki Wrocławskiej.
9. Kowalski, R. (1989). *Logika w rozwiązywaniu zadań*. WTN.
10. Kwaśnicka, H. (1998). Algorytmy genetyczne w uczeniu się maszyn. *Prace Naukowe Akademii Ekonomicznej we Wrocławiu, no. 787,* pp. 148-169.
11. Martin, R. (2014). *Czysty kod. Podręcznik dobrego programisty*. Helion.
12. McCabe, F., Clark, K. (1998). *Micro-Prolog*. WNT.
13. Mulawka, J. (1997). *Systemy ekspertowe*. WNT.
14. Nieberliński, A. (2000). *Regułowe systemy ekspertowe*. Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego. ISBN: 8386644370.
15. Pająk, E. (2006). *Zarządzanie produkcją. Produkt, technologia, organizacja*. PWN.

16. Shortliffe, E.H., Buchanan, B.G. (1984). *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Addison Weseley.
17. Zieliński, J. (2000). *Inteligentne systemy w zarządzaniu*. PWN.
18. Zoleński, W. (2008). *Metoda budowy informatycznego systemu wczesnego ostrzegania w zarzadzaniu w przedsiębiorstwie*. Rozprawa doktorska. Wydawnictwo Politechniki Śląskiej.
19. Zoleński, W. (2014). Koncepcja systemu ekspertowego wspomagającego pozyskiwanie wiedzy. *Zeszyty Naukowe Politechniki Śląskiej, no. 70*.