

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Donatas VITKUS

RESEARCH ON AUTOMATED
KNOWLEDGE BASE GENERATION
METHODS OF INFORMATION SECURITY
RISK ANALYSIS EXPERT SYSTEMS

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
INFORMATICS ENGINEERING (T 007)

Vilnius, 2022

Doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2017–2022.

Supervisor

Assoc. Prof. Dr Nikolaj GORANIN (Vilnius Gediminas Technical University, Informatics Engineering – T 007).

The Dissertation Defense Council of Scientific Field of Informatics Engineering of Vilnius Gediminas Technical University:

Chairman

Prof. Dr Dalius MAŽEIKA (Vilnius Gediminas Technical University, Informatics Engineering – T 007).

Members:

Prof. Dr Rimantas BUTLERIS (Kaunas University of Technology, Informatics Engineering – T 007),

Dr Robertas DAMAŠEVIČIUS (Kaunas University of Technology, Informatics Engineering – T 007),

Prof. Dr Habil. Gintautas DZEMYDA (Vilnius University, Informatics Engineering – T 007),

Assoc. Prof. Dr Marko HÖLBL (University of Maribor, Slovenia, Informatics – N 009).

The dissertation will be defended at the public meeting of the Dissertation Defense Council of Informatics Engineering in the Senate Hall of Vilnius Gediminas Technical University at **1 p. m. on 27 May 2022.**

Address: Saulėtekio al. 11, LT-10223 Vilnius, Lithuania.

Tel.: +370 5 274 4956; fax +370 5 270 0112; e-mail: doktor@vilniustech.lt

A notification on the intend defending of the dissertation was sent on 26 April 2022. A copy of the doctoral dissertation is available for review at Vilnius Gediminas Technical University repository <http://dspace.vgtu.lt>, at the library of Vilnius Gediminas Technical University (Saulėtekio al. 14, LT-10223 Vilnius, Lithuania) and at the library of Kaunas University of Technology (K. Donelaičio st. 20, LT-44239 Kaunas, Lithuania).

Vilnius Gediminas Technical University book No 2022-014-M

doi:10.20334/2022-014-M

© Vilnius Gediminas Technical University, 2022

© Donatas Vitkus, 2022

donatas.vitkus@gmail.com

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Donatas VITKUS

INFORMACIJOS SAUGOS RIZIKOS
ANALIZĖS EKSPERTINIŲ SISTEMŲ ŽINIŲ
BAZĖS AUTOMATIZUOTO GENERAVIMO
METODŲ TYRIMAS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
INFORMATIKOS INŽINERIJA (T 007)

Vilnius, 2022

Disertacija rengta 2017–2022 metais Vilniaus Gedimino technikos universitete.

Vadovas

doc. dr. Nikolaj GORANIN (Vilniaus Gedimino technikos universitetas, informatikos inžinerija – T 007).

Vilniaus Gedimino technikos universiteto Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:

Pirmininkas

prof. dr. Dalius MAŽEIKAS (Vilniaus Gedimino technikos universitetas, informatikos inžinerija – T 007).

Nariai:

prof. dr. Rimantas BUTLERIS (Kauno technologijos universitetas, informatikos inžinerija – T 007),

dr. Robertas DAMAŠEVIČIUS (Kauno technologijos universitetas, informatikos inžinerija – T 007),

prof. habil. Dr. Gintautas DZEMYDA (Vilniaus universitetas, informatikos inžinerija – T 007),

doc. dr. Marko HÖLBL (Mariboro universitetas, Slovėnija, informatika – N 009).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje **2022 m. gegužės 27 d. 13 val.** Vilniaus Gedimino technikos universiteto senato posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuva.

Tel.: (8 5) 274 4956; faksas (8 5) 270 0112; el. paštas doktor@vilniustech.lt

Pranešimai apie numatomą ginti disertaciją išsiųsti 2022 m. balandžio 26 d.

Disertaciją galima peržiūrėti Vilniaus Gedimino technikos universiteto talpykloje <http://dspace.vgtu.lt> ir Vilniaus Gedimino technikos universiteto (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva) bei Kauno technologijos universiteto (K. Donelaičio g. 20, LT-44239 Kaunas, Lietuva) bibliotekose.

Abstract

In recent years, fast digitalization and the increasing use of information technologies have increased the efficiency of companies and posed new challenges for information security. Changing information technologies (IT) are becoming more accessible to businesses of all sizes; increasingly more cloud computing services are being developed. It would be difficult to find an organization nowadays that has no digital information to secure.

Information security risk assessment can help identify vulnerabilities and risks on time and select risk mitigation measures. The main challenge for information security risk assessment is having the personnel competent to perform such assessments. This problem is particular to small and medium-sized businesses. Such issues can be addressed using expert systems (ESs) that enable information security risk assessments without information security experts. However, the main problem with the use of ES for information security risk assessment is the updating of the ES knowledge base.

The knowledge base needs to be updated regularly and frequently due to ever-changing threats. Updating the knowledge base is a costly and complex process that requires the involvement of experts. To date, most knowledge bases are created and updated manually, so there is a need to automate this process, ensuring the knowledge base is relevant and the process is cheaper and faster. It is also important to reuse the existing sources of information security knowledge with reliable and relevant information.

The dissertation consists of an introduction, five main chapters, and general conclusions. The first chapter introduces ES for information security risk assessment, discusses the existing methods of forming and updating the ES knowledge base, and identifies potential sources of information security knowledge. The second chapter presents an approach to transforming ontologies into ES knowledge base rules. The third chapter presents a method for automatically transferring information from website (WEB) sources to the ES knowledge base. The fourth chapter presents an approach to transforming attack trees into ES knowledge base rules, which allows the ES knowledge base to be expanded with the risks of cyber-attacks. The proposed methods are implemented and tested by importing the developed rules into Java Expert System Shell (JESS)-based ES. The fifth chapter describes the developed ES prototype for small and medium-sized enterprises (SMEs), whose knowledge base is created by applying the proposed automation methods.

The performed experiments and analysis have demonstrated that the proposed methods allow the automated formation of the ES knowledge base and its use in information security risk assessment.

Reziumė

Pastaraisiais metais vykstanti greita įmonių skaitmenizacija ir informacinių technologijų naudojimo didėjimas didina ne tik įmonių veiklos efektyvumą, bet taip pat kelia ir naujų iššūkių, susijusių su informacijos sauga. Informacinės technologijos (IT) keičiasi ir tampa prieinamesnės įvairaus dydžio įmonėms, kuriama vis daugiau debesų kompiuterijos paslaugų. Šiandien jau būtų sunku surasti organizaciją, kuri neturėtų skaitmeninės informacijos, kurią reikia saugoti.

Informacijos saugos rizikos vertinimas gali padėti laiku nustatyti silpnąsias vietas ir jų keliamą pavojų bei parinkti rizikas mažinančias priemones. Pagrindinis informacijos saugos rizikos vertinimo iššūkis – kompetentingas personalas, galintis atlikti tokius vertinimus. Mažoms ir vidutinėms įmonėms tai tampa problema. Tokioms problemoms spręsti siūloma naudoti ekspertines sistemas (ES), kurios leidžia atlikti informacijos saugos rizikos vertinimus neturint informacijos saugos ekspertų. Tačiau pagrindinė ES, skirtų informacijos saugos rizikai vertinti, naudojimo problema yra ES žinių bazės atnaujinimas.

Žinių bazę dėl nuolat kintančių grėsmių reikia atnaujinti reguliariai ir dažnai. Žinių bazės atnaujinimas – brangus ir sudėtingas procesas, reikalaujantis ekspertų įsitraukimo. Iki šiol didžioji dalis žinių bazių kuriama ir atnaujinama rankiniu būdu, todėl atsiranda poreikis automatizuoti šį procesą, taip užtikrinant, kad žinių bazė yra aktuali, o procesas taptų pigesnis ir greitesnis. Taip pat aktualu panaudoti jau egzistuojančius informacijos saugos žinių šaltinius su patikima ir aktualia informacija.

Disertaciją sudaro įvadas, penki pagrindiniai skyriai ir bendrosios išvados. Pirmajame skyriuje pristatomos informacijos saugos rizikų vertinimo ES, aptariami egzistuojantys ES žinių bazės formavimo ir atnaujinimo metodai, identifikuojami potencialūs informacijos saugos žinių šaltiniai. Antrajame skyriuje pristatomas ontologijų transformavimo į ES žinių bazės taisykles metodas. Trečiajame skyriuje pristatomas metodas, skirtas į ES žinių bazę automatizuotai perkelti informaciją iš WEB šaltinių. Ketvirtajame skyriuje pristatomas atakų medžių transformavimo į ES žinių bazės taisykles metodas, kuris leidžia ES žinių bazę praplėsti kibernetinių atakų rizikomis. Pateikti metodai yra realizuojami ir išbandomi importuojant sukurtas taisykles į *Java Expert System Shell* (JESS) pagrindu sukurtas ES. Penktajame skyriuje aprašomas sukurtas ES prototipas, skirtas smulkaus ir vidutinio dydžio įmonėms (SME), jo žinių bazę sukurta pritaikant siūlomus automatizavimo metodus.

Ekspperimentinio tyrimo metu nustatyta, kad siūlomi metodai leidžia automatizuoti ES žinių bazės formavimą ir automatizuotai suformuotą žinių bazę taikyti informacijos saugos rizikai vertinti.

Notations

Symbols

A – the area of attacks
AT – the area of attack trees
CLP – CLIPS rules
ES – expert systems
F – the area of facts
FG – the area of knowledge base facts and aims
G – the area of attack aims
KB – knowledge bases
N – the area of attack tree leaves
NN – node template
NR – the area of attack tree elements
O – ontology
o – RDF object
p – RDF predicate
R – the final set of RIF rules
R – the area of root nodes
RDF (O) – RDF triples of ontology *O*
s – RDF subject

S – the source of the attack tree

T – the area of RDF triplets

TT – the attack tree template

Abbreviations

ACME – a company making everything

ADTool – attack-defense tree tool

AI – artificial intelligence

AT – attack tree

ATES – attack trees to expert system

ATMS – assumption-based truth maintenance system

ATT&CK – adversarial tactics, techniques, and common knowledge

BPM – business process model

C&C – command & control

CBR – case-based reasoning

COBIT – control objectives for information and related technology

CPE – product dictionary

CVE – common vulnerabilities and exposures

CVSS – common vulnerability scoring system

CWE – common weakness enumeration

DDOS – distributed denial of service

DOS – denial of service

DSS – decision support systems

ES – expert systems

HTML – hypertext markup language

HTTP – hypertext transport protocol

ISO/IEC – international organization for standardization/international electrotechnical commission

ISRM – information security risk management

IT – information technologies

JESS – java expert system shell

MRM – matrix representation and mapping

NIST – national institute of standards and technology

NVD – national vulnerability database

RIF – rule interchange format

SCADA – supervisory control and data acquisition

SME – small and medium-sized enterprises

UATS – universal metamodel for attack trees

UML – unified modeling language

W3C – worldwide web consortium

WEB – website

XML – extensible markup language

Contents

INTRODUCTION	1
Problem Formulation.....	1
Relevance of the Thesis.....	2
Research Object.....	3
Aim of the Thesis	3
Tasks of the Thesis	3
Research Methodology	4
Scientific Novelty of the Thesis	4
Practical Value of the Research Findings.....	5
Defended Statements.....	5
Approval of the Research Findings	6
The Structure of the Dissertation.....	6
 1. ANALYSIS OF EXPERT SYSTEMS, DEDICATED TO INFORMATION SECURITY RISK ANALYSIS AND THEIR KNOWLEDGE BASE FORMATION METHODS.....	 7
1.1. Expert Systems for Information Security Risk Analysis	8
1.1.1. Overview of Information Security Risk Management Process	8
1.1.2. General Definition of an Expert System	10
1.1.3. Review of Expert System Shells.....	16

1.1.4. Rule Inference Algorithms.....	18
1.1.5. Review of Expert Systems Use for Risk Analysis	20
1.2. Expert Systems' Knowledge Base Formation Methods	26
1.2.1. Review of Existing Manual Expert Systems' Knowledge Base Formation Methods	27
1.2.2. Review of Automated Expert Systems' Knowledge Base Rules Generation Methods	30
1.2.3. Review of Automatic Expert Systems' Knowledge Base Rules Generation Methods	31
1.3. Potential Sources for Expert System's Knowledge Base Formation Automation.....	36
1.3.1. Ontology as a Source for Knowledge Base	39
1.3.2. Web Data as a Source for Knowledge Base	41
1.3.3. Attack Trees as a Source of Risk Analysis Information	44
1.4. Related Research in Lithuania	47
1.5. Conclusions of Chapter 1 and the Formulation of Tasks.....	49
 2. AUTOMATIC APPROACH OF ONTOLOGY TRANSFORMATION INTO EXPERT SYSTEM'S KNOWLEDGE BASE	51
2.1. Ontology Integration via Rule Interchange Format	51
2.1.1. Rule Interchange Format	52
2.1.2. Web Ontology Language Integration with Rule Interchange Format.....	53
2.1.3. Transforming Web Ontology Language (OWL 2 RL) to Rule Interchange Format (RIF CORE)	54
2.2. Method for Transforming an Ontology into Rules	55
2.3. Method Implementation for Transforming Ontologies into Rules of an Expert System's Knowledge Base.....	56
2.3.1. Design of an Ontology Transformation System	56
2.3.2. Software for Method Implementation.....	60
2.3.3. Example of Transformation	65
2.4. Method Experiments and Verification.....	68
2.5. Conclusions of Chapter 2	70
 3. INTEGRATION OF DATA ON REGIONAL MALWARE DISTRIBUTION INTO THE EXPERT SYSTEM'S KNOWLEDGE BASE FOR GEOGRAPHICALLY ADAPTED RISK EVALUATION	73
3.1. Integration of Data on Regional Malware Distribution from the Cyberthreat Real-Time Map into the Knowledge Base	74
3.1.1. Acquisition of Data from Websites.....	75
3.1.2. Knowledge Base Formation and Import into an Expert System	76
3.2. Method Experiments and Risk Analysis Based on Regional Malware Distribution	80
3.3. Conclusions of Chapter 3	82

4. METHOD FOR ATTACK TREE DATA TRANSFORMATION AND IMPORT INTO IT RISK ANALYSIS EXPERT SYSTEMS	85
4.1. Automation of Converting Information from Attack Trees into Expert System's Knowledge Base Rules	86
4.1.1. Attack Tree Preparation for Transformation.....	87
4.1.2. Converting UATS to Expert System Rules.....	88
4.2. Proposed Method Implementation.....	95
4.3. Proposed Method Experiments and Results	97
4.4. Conclusions of Chapter 4	103
5. PROTOTYPE EXPERT SYSTEM FOR INFORMATION SECURITY RISK ANALYSIS IN SMALL-MEDIUM ENTERPRISES	105
5.1. Expert System's Design for Information Security Risk Analysis.....	106
5.1.1. Requirements for the Expert System	106
5.1.2. Architecture of the Developed Expert System.....	107
5.1.3. Implementation of the Expert System.....	108
5.2. Developed Expert System	112
5.3. Experiments and Validation	115
5.4. Conclusions of Chapter 5	123
GENERAL CONCLUSIONS	125
REFERENCES	127
LIST OF SCIENTIFIC PUBLICATIONS BY THE AUTHOR ON THE TOPIC OF THE DISSERTATION.....	145
SUMMARY IN LITHUANIAN	147
ANNEXES ¹	165
Annex A. Fixed Rules for Method of Transforming an Ontology into Rules	165
Annex B. Examples of the developed tools.....	179
Annex C. Expert Knowledge Verification Questionnaire	183
Annex D. Declaration of Academic Integrity.....	189
Annex E. Copies of Scientific Publications by the Author on the Topic of the Dissertation	190

¹ The annexes are supplied in the enclosed compact disc.

Introduction

Problem Formulation

Information security is commonly defined as the properties or characteristics that secure information should have. Information security should be ensured by a complex of technical and organizational measures, and information security risk analysis is seen as a core organizational process for decision-making in information security management. According to the European General Data Protection Regulation, a risk-based approach to data protection is embraced (Maldoff, 2017).

Enterprises have two ways of ensuring their information security: employ a specialist or outsource the risk analysis service. The price of both choices is relatively high. Some researchers (Janulevičius & Čenys, 2014; Kailay & Jarratt, 1995; Kozhakhmet et al., 2012) propose expert systems as the solution for risk analysis automation since the use of human experts is expensive and time-consuming. Expert systems play a crucial role in taking the knowledge from a security expert, expressed as rules, and allowing it to be shared effortlessly. By design, expert systems need a knowledge base, which must be up-to-date and high-quality. Sometimes, it becomes a problem since high-quality knowledge is expensive. Manual creation of knowledge bases is expensive and cannot ensure stable information renewal.

Therefore, the knowledge base automation process is crucial. On the other hand, many sources of systematic information can be used for knowledge base creation, reducing the process expenses at least partially.

Automated generation of the expert system's knowledge base was proposed to solve the problem related to the formation and renewal of the expert system's knowledge base. Transforming security ontologies into rules of the expert system's knowledge base allows the creation of high-quality static information knowledge bases for expert systems. The method based on acquiring data from trusted and open websites providing current information on newly discovered threats enables region-based security risk analysis. Knowledge base rules, created from attack trees, allow specialized, more detailed, and complex security risk analysis, targeting different information systems.

Relevance of the Thesis

Several authors (Lashkarara, 2017; Ralston et al., 2007; Shang et al., 2019) agree that today information is the critical business part despite the company's size. Even the smallest company has information and information systems to be secured (Stowell, 1993). However, as enterprises become more complex, integrated, and connected to third parties, the budget for security and controls quickly reaches its limits (Schmittling, 2010).

Big and modern companies can afford risk-based information security management (Agrawal, 2017) and experts for security risk analysis. However, most companies have no particular knowledge of security risk management (Gerber et al., 2016; Kozubíková et al., 2015). Therefore, security risk analysis is becoming an actual problem for SMEs and smaller companies with no required resources (Alcántara & Melgar, 2016; Gupta & Hammond, 2005). In many cases, they misunderstand the value of risk analysis and perform it irregularly. Consequently, this limitation widens the technological and competitive gap between SMEs and large enterprises, making the former uncompetitive.

Therefore, the security risk analysis process must be automated by introducing expert systems or decision support systems, which could help SMEs perform an information security risk analysis without any special knowledge and security experts. Also, the security risk analysis process must become faster and cheaper (Powell, 1996), i.e., affordable for SMEs, as currently it is implemented using malware management via antiviruses.

Presently, sufficient information is collected and presented on the Internet in different forms (Gero & Maher, 2013; Obrst et al., 2014), including security information, such as standards, best practices, and security ontologies (Blanco et al., 2008). A good-quality knowledge base requires obtaining knowledge, e.g., by

combining extensive literature review and expert mining (Kapse et al., 2018) rather than using one source.

The idea to automate knowledge base formation is relatively new; still, some research in this area has already been done, and the currently proposed methods have demonstrated the possibility and benefits of such an approach. By developing a process for risk assessment using expert systems, the analysis can be made more affordable for small and medium enterprises (SMEs) as a business category, simultaneously increasing its accuracy (Gadyatskaya, Harpes, et al., 2016; Janulevičius & Čenys, 2014) for everyone, and especially SMEs failing to afford high-quality security experts.

Research Object

The object of the present study is methods for expert systems, dedicated to information security risk analysis, knowledge base generation automation.

Aim of the Thesis

The aim of this research is to improve automated information security risk analysis expert systems knowledge base generation methods to allow reducing the segregation in area of information security management for small and medium-sized enterprises using existing information security sources.

Tasks of the Thesis

To achieve the aim of the thesis, the following tasks had to be accomplished:

1. To review existing expert systems, dedicated for information security risk analysis, and their knowledge base formation methods, identify potential information sources and provide their application areas.
2. To propose methods for improving automated information security risk analysis expert system knowledge base generation using existing information security sources.
3. To perform experimental verification and validation of proposed automated knowledge base generation methods for expert systems, evaluate the obtained results, and define the objectives for future work.

Research Methodology

The following research methods were chosen for investigation:

- Action: a theoretical (analysis and synthesis) study has been performed to improve strategies aimed at finding a solution to the problem.
- Classification: strengths, weaknesses, and existing gaps presented in the literature were summarized, aiming to recognize and understand the dissertation's research object.
- Experience: the solution to the problem has been found by intuition and experience.
- Experiment: the hypothesis has been tested practically.
- Statistics: conclusions have been drawn by collecting, analyzing, and explaining the statistical data.

Scientific Novelty of the Thesis

The scientific novelty of this study is specified as follows:

1. A novel method is proposed to automate the transformation of existing information security ontologies into expert system knowledge base rules and is used to perform information security risk analysis. Unlike other methods, it does not integrate ontology directly but utilizes the transformation of ontology elements into expert system rules.
2. A novel method is proposed for the generation of a knowledge base of automated expert systems based on the integration of data on regional malware distribution from a real-time web map providing current information on newly discovered threats. The use of real-time map information allows for more precise region-adopted risk calculation.
3. A novel method is proposed to convert attack trees to a format usable by expert systems for utilizing existing attack tree repositories, storing information on attack likelihood and cost to facilitate information and information security risk analysis.

Practical Value of the Research Findings

The achieved results are important from the practical viewpoint of the dynamic and rapidly changing decision-making process of information security management, which also implements a compulsory requirement stipulated in regulating documents.

The proposed knowledge base generation methods for expert systems used in information security risk analysis that use existing information security ontologies, malware distribution, and attack trees' information demonstrate an interdisciplinary approach, combining both informatics engineering and managerial methods aimed at solving risk-based information security tasks.

The developed tools for automated transformation ontologies, attack trees, and malware distribution into rules of an expert system knowledge base and integration into information security risk analysis expert systems can be directly used by companies, especially small-medium sized enterprises, for minimizing time and cost expenses while performing information security risk analysis without having any special information security knowledge.

The developed expert system prototype has demonstrated the practical applicability of an automated knowledge base in the risk analysis process.

Defended Statements

The defended statements of this doctoral thesis are as follows:

1. The transformation of various information sources with different characteristics into rules of expert systems and their inclusion in the knowledge base significantly increases the amount of relevant information operated by the expert system.
2. The automatically formed rules of expert systems integrated into the knowledge base complement it with the information required for the performance of information security risk analysis and are consistent with the manually formed rules.
3. Due to the complexity of information security risk analysis, there is currently no single source that would be sufficient to form the knowledge base of the expert system. Different sources of information complement the knowledge base of the expert system with specific information needed for information security risk management.

Approval of the Research Findings

The results of the dissertation were published in three scientific publications in reviewed scientific journals indexed in Clarivate Analytics (also referred to as Thomson Reuters) Science Citation Index. The author has also made four presentations at scientific conferences. Two of them were international (one was hosted abroad), and two were local:

- The 24th International Nordic Conference on Secure IT Systems (NordSec 2019), Aalborg, Denmark.
- The 11th International Workshop Data Analysis Methods for Software Systems (DAMSS 2019), Druskininkai, Lithuania.
- The 22nd Junior Researchers Conference “Science – Future of Lithuania” on IT Security and Information Systems, 2019, Vilnius, Lithuania.
- The 23rd Junior Researchers Conference “Science – Future of Lithuania” on IT Security and Information Systems, 2020, Vilnius, Lithuania.

The Structure of the Dissertation

The dissertation is structured around five main chapters, general conclusions, references, a list of the author’s publications, and annexes. The total scope of the dissertation is 146 pages without annexes. The text contains 54 pictures and 12 tables. Two hundred twenty references were used in the dissertation text.

Analysis of Expert Systems, Dedicated to Information Security Risk Analysis and Their Knowledge Base Formation Methods

This chapter reviews expert systems for information security risk analysis and automated generation of their knowledge bases. It begins by explaining the process of information security risk management, which outlines its key elements. Next, the text provides the basic architecture of the expert system and its description and provides an analysis of existing expert systems dedicated to information security risk analysis. The analysis of the existing knowledge base generation methods presents different knowledge base generation techniques. Suitable potential information sources are analyzed to evaluate the possibilities of automated knowledge base generation. The chapter reasons the selection of three different sources used in later experiments. It concludes by analyzing related research by Lithuanian authors and summarizing the literature review findings.

The analysis presented in this chapter was published in Vitkus et al. (2019), Vitkus, Jezukeviciute, et al. (2020), Vitkus, Salter, et al. (2020).

1.1. Expert Systems for Information Security Risk Analysis

Expert systems (ESs) are a branch of applied artificial intelligence developed by the Artificial Intelligence (AI) community in the mid-1960s. The main idea of an expert system is to simulate the tasks solving process of a real human expert who has specific knowledge used based on the situational analysis for decision making (Liao, 2005).

The 60-year-old concept of expert systems is still used. Current ESs are widely used for engineering and manufacturing applications. There were some attempts to use expert systems for automating a particular part of security risk analysis (Atymtayeva et al., 2014). The main reason for such attempts is to make the risk analysis process faster and cheaper (Willcocks, 2013). Expert systems for information security risk analysis are based on questions about information security policy implementation, e.g., password strength, physical access, personal knowledge, and behavior (Sharda et al., 2014).

1.1.1. Overview of Information Security Risk Management Process

Nowadays, information plays a critical role in an enterprise. Every business, even the smallest one, has information that must be secured. Almost all processes in today's organization are based on information technologies. Information security risk management can help enterprises reduce risks to an acceptable level.

Information security risk management (ISRM) is a set of processes, supporting tools, and staff for organizations to proactively identify threats and vulnerabilities, assess potential exposure, impact and likelihood, and ultimately ascertain whether their security controls are appropriate for the faced threats (Young, 2020). There are several different ISRM frameworks, methodologies, and standards.

ISO 31000² – an international risk management standard – and ISO 27005³ – a risk management standard related to international information security – support the related standards of the ISO 27000 family, such as ISMS standard 27001. Another well-known framework is the National Institute of Standards and Technology (NIST) SP 800-30⁴. It provides risk assessment guidelines for organization and government information systems and complements NIST SP 800-39⁵, which

² <https://www.iso.org/iso-31000-risk-management.html>

³ <https://www.iso.org/standard/75281.html>

⁴ <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf>

⁵ <https://csrc.nist.gov/publications/detail/sp/800-39/final>

is dedicated to information security risk management. OWASP Risk Rating Methodology⁶ is dedicated to information security risk management. It helps to estimate the severity of security risks to the business and make an informed decision about what to do about those risks.

ISRM process has several parts. Differences depend on frameworks or standards, but the main parts remain the same. The risk management process, according to ISO 31000, includes these main steps:

- Communication and consultation with external and internal stakeholders should occur during all stages of the risk management process.
- Establishing the context addresses both company external and internal factors, the role of the risk management process within the company, and basic criteria used to evaluate risks.
- Risk assessment involves three main sub-processes, namely, risk identification, analysis, and evaluation. It is the most important step in security risk management (Albakri et al., 2014).
- Risk identification covers the identification of assets, threats, existing controls, vulnerabilities, and consequences.
- Risk analysis identifies a deeper understanding of previously developed risks. It generates the necessary information for a correct evaluation of the risk. It includes the root cause analysis, likelihood identification, and qualifying or quantifying the risk level.
- Risk evaluation is based on the information gathered in the risk analysis, and decisions are made about risks that need treatment and the priority of such treatment.
- Risk treatment involves assessing different treatments, the resulting residual risks, and deciding whether additional risk treatments are necessary to achieve the intended risk reduction.
- Monitoring and review definition states that the identified risks, including the identification of emerging risks, must be monitored and reviewed.

The risk management process is open-ended. The simplified formal risk management process is depicted in Fig. 1.1.

⁶ https://owasp.org/www-community/OWASP_Risk_Rating_Methodology



Fig. 1.1. Formal risk management process

As provided in the picture, the main idea of the risk management cycle is continuous improvement. Risk assessment (risk identification, analysis, and evaluation) becomes an input to risk response (or treatment). Risk responses do not close the risk management cycle but open it from the beginning. Risk response outputs become risk identification inputs. It means that risk identification should include the last risk assessment and risk response results, and there is a need to reassess them.

1.1.2. General Definition of an Expert System

Feigenbaum is one of the ES pioneers. He said that an expert system is an intellectual computer program, which uses knowledge and inference procedures to solve tasks that are very complex and require extensive human expertise (Feigenbaum, 1982). It is mainly a knowledge-based program that provides expert quality solutions to the problem in a specific domain. It is impossible to get solutions from human experts every time for every specific problem as a human expert may not be available everywhere. A solution to such problems is given by knowledge-based computer systems (Mankad, 2018).

Expert systems are among the first AI implementations (Negnevitsky, 2005). Medical and engineering software developers used ESs to implement the decision-making process using symbolic but not numerical values (Harrison et al., 1993).

An expert system suggested by Edward Feigenbaum (Feigenbaum, 1982) has several parts. A basic structure of an expert system is shown in Fig. 1.2.

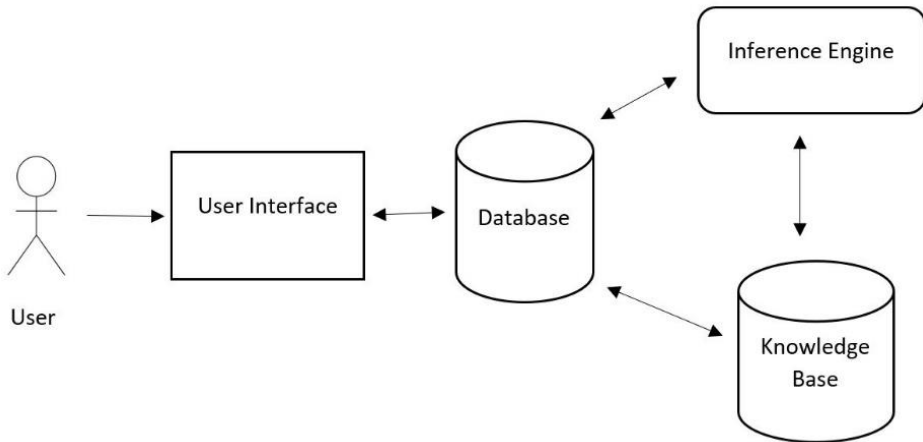


Fig. 1.2. Basic Structure of an Expert System

The user interface allows the user to enter rules and facts about a situation and ask the system questions. It responds to user requests and supports all communication between the system and the user.

The knowledge base constitutes the problem-solving rules, facts, or intuition that a human expert might use in solving problems in each problem domain. The knowledge base is usually stored in the form of *if-then* rules. It is domain-specific and contains heuristics useful for the problem domain.

The domain database contains facts about the expert system's subject. It contains short-term data, inference states, and a history of solutions.

The inference engine is the control mechanism that organizes the problem data and searches through the knowledge base for applicable rules. It uses the information provided by the knowledge base and the user to infer new facts (Mankad, 2018).

Expert systems have evolved and improved, their architecture has changed, and now several categories can be distinguished. The main ones are rule-based systems, knowledge-based systems, neural networks, fuzzy ES, object-oriented methodology, case-based reasoning, and model-based systems, together with their applications for different research and problem domains (Liao, 2005).

A rule-based ES is defined as containing information obtained from a human expert and representing that information in the form of rules, such as IF–THEN. These *if-then* rule statements are used to formulate the conditional statements that

comprise the complete knowledge base. A single *if-then* rule assumes the form “*if x is A, then y is B*,” and the if-part of the rule “*x is A*” is called the antecedent or premise, while the then-part of the rule “*y is B*” is called the consequent or conclusion.

There are two broad kinds of inference engines used in rule-based systems: forward chaining and backward chaining systems (Abraham, 2005). Fig. 1.3 shows an example of forward chaining and backward chaining (Hill, 2003).

Rules: IF $a=1$ & $b=2$ THEN $C=3$

IF $C=3$ THEN $d=4$

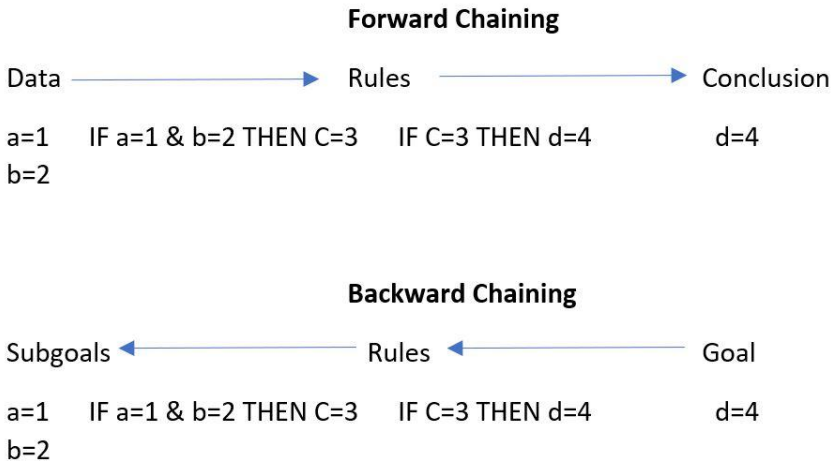


Fig. 1.3. Forward chaining vs. backward chaining

In Fig. 1.3, a , b , C , and d are linguistic variables of production rules. A forward-chaining system processes the initial facts first and keeps using the rules to draw new conclusions given those facts. A backward-chaining system first processes the hypothesis (or solution/goal) to be reached and keeps looking for rules that would allow concluding that hypothesis. As the processing progresses, new subgoals are also set for validation. Forward chaining systems are primarily data-driven, while backward chaining systems are goal-driven.

There is a wide application range of rule-based ES: medicine, production planning, advisory system, teaching, system development, knowledge verification, and many others.

A knowledge-based system is defined as human-centered. A typical structure of a knowledge-based expert system (Alder et al., 2014) is shown in Fig. 1.4.

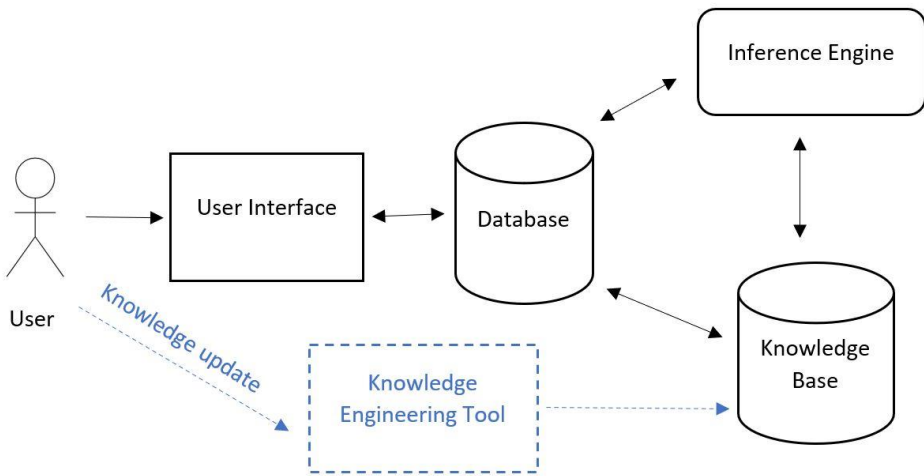


Fig. 1.4. Typical structure of a knowledge-based expert system

Such systems have four main components: a knowledge base, an inference engine, a knowledge engineering tool, and a specific user interface (Jasbir S. Dhaliwal & Benbasat, 1996). This term also includes all the organizational information technology applications that may prove helpful in managing the knowledge assets of an organization, such as ES, rule-based engines, groupware, and database management (Caldas, 2003).

The application of knowledge-based systems is as wide as rule-based ES.

Decision support systems (DSS) are designed to ensure more precise decision-making by effectively using timely and appropriate data, information, and knowledge management (Chung et al., 2016). DSS are similar to ES because they are computer-based programs and technologies intended to make routine decisions and monitor and control processes typically done by humans (Millet & Fenwick, 2005). Compared to ES, DSS are more autonomous but not as widely applied.

Artificial neural networks can be viewed as supporting at least in two types of DSS, i.e., data-driven and model-driven. First, neural networks can be employed as data analysis tools for forecasting and prediction based on historical data in a data-driven DSS. Second, neural networks also can be viewed as a class of quantitative models to be used in a model-driven DSS (Delen & Sharda, 2008). An artificial neural network is a model that emulates a biological neural network. This concept is used to implement software simulations for the massively parallel processes that involve processing elements interconnected in network architecture (Turban, J. E. A. Efrain, 2018). A recurrent network has three layers: an input layer, a hidden layer, and an output layer. The size of the input layer depends on

the number of inputs and the number of outputs used to denote the states. The backpropagation algorithm is used as the learning algorithm for this network, and the sigmoid function is used as the transfer function in the hidden layer. A serial adder and a sequential decoder are used for the implementation of the sequential machine (Wei et al., 2001).

Some of the applications that are implemented by neural networks are an alarm processing system, a diagnostic system, malware detection, security assessment, Distributed Denial of Service (DDoS) attack detection and mitigation, and risk evaluation.

Fuzzy expert systems are developed using the fuzzy logic method, which deals with uncertainty. This technique uses the mathematical theory of fuzzy sets and simulates the process of normal human reasoning by allowing the computer to behave less precisely and logically than conventional computers. The decision-making process is not always true or false, black or white, 1 or 0. It often involves gray areas and the term “may be” (Berredjem & Benidir, 2018).

Fig. 1.5 design is based on Abraham (2005) and illustrates the basic architecture of a fuzzy expert system.

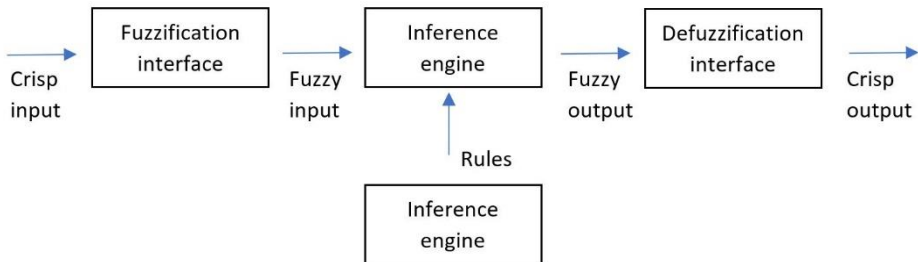


Fig. 1.5. Basic architecture of a fuzzy expert system

The main components of a fuzzy expert system are a fuzzification interface, a fuzzy rule base (knowledge base), an inference engine (decision-making logic), and a defuzzification interface. The input variables are fuzzified, whereby the membership functions defined on the input variables are applied to their actual values to determine the degree of truth for each rule antecedent. According to Mamdani, in a fuzzy inference system (Mamdani & Assilian, 1975), the rule antecedents and consequents are defined by fuzzy sets.

Fuzzy ESs are used in power load forecasting (Kiartzis et al., 2000), online scheduling (DENG et al., 2018), diabetes diagnosis (Abdullah et al., 2018; Mujawar & Jadhav, 2019), cyber-attack detection (Khosravi & Fereidunian, 2019), risk analysis (Nourian et al., 2019), security intelligence modeling (Korać & Simić, 2017), and malware detection (Paulo et al., 2020).

An object-based expert system combines object data and specific procedures that operate on this data, where the object combines data and program code. Instead of passing data to procedures, programs send a message for an object to perform a procedure that is already embedded in it. Then, the same message may be sent to many different objects, but each will implement that message differently. An object's data are encapsulated from other parts of the system, so each object is an independent software building block that can be used in many different systems without changing the program code (Liao, 2005).

This system differs from a rule-based expert system by the knowledge representation methodology. It relates an object or item to various facts or values (Abraham, 2005). Some applications implemented by an object-based ES include industry diagnosis, knowledge learning, manufacturing information network, power system maintenance, knowledge engineering, syntactic programming, and knowledge representation.

Case-based Reasoning (CBR) is an approach to solving problems based on solutions of similar past cases (Aamodt & Plaza, 1994). In CBR, descriptions of human specialist experiences, represented as cases, are stored in a database for later retrieval when the user encounters a new case with similar parameters. The system searches for stored cases with characteristics particular to the new problem, finds the closest fit, and applies the available solutions to the new case. Successful solutions are tagged to the new case, and both are stored together with other cases in the knowledge base. Unsuccessful solutions are also appended to the case base, along with explanations as to why the solutions did not work (Kolodner, 2014).

Some of the applications implemented by CBR include manufacturing process design, knowledge management, power system restoration training, ultrasonic inspection, medical planning, medical application, fault diagnosis, e-learning, and knowledge modeling (Liao, 2005).

Model-based systems have been developed as a way to exploit existing knowledge of the domain principles and to extend the limited flexibility that is often found in classical expert systems. The model-based approach emphasizes the representation of the structure and function of the modeled system and reasoning with respect to that structure and function (Kunz et al., 1989).

The central component of such architecture is an assumption-based truth maintenance system. An Assumption-based Truth Maintenance System (ATMS) is an inference engine that enables a problem solver to reason about multiple possible worlds or situations. Each possible world describes a specific set of circumstances, a crime scenario in this application, under which certain events and states are true, and other events and states are false. What is true in one possible world may be false in another. The task of the ATMS is to maintain what is true in each possible world (Schafer & Keppens, 2004).

Modeling is used in process control, medical analysis, management decision making, software evaluation, medical system validation, assembly task planning and simulation, transport terminal design, project allocation, and endometrial hyperplasia classification (Liao, 2005).

In summary, rule-based and knowledge-based ESs are the most classical expert systems. Other types are more like sub-categories of expert systems for solving specific ES problems. This thesis focused on rule-based expert systems' knowledge bases. The main reason behind this choice is that rule-based ESs are the most popular and universal, applicable in various areas, and include the information security risk assessment.

1.1.3. Review of Expert System Shells

Most expert systems are developed using specialized software called a shell, especially when it comes to rule-based expert systems. A shell allows the user just to create the knowledge base in the form of simple rules without paying attention to knowledge interpretation engine development, thus minimizing the ES development time (Negnevitsky, 2005).

A shell refers to the software module containing an interface, an inference engine, and a structured skeleton of a knowledge base (in its empty state) with appropriate knowledge representation facilities. In layman's terms, it can be viewed as an empty bowl yet to be filled with expert knowledge elements that, along with the inference engine, may be used to process user requests to generate solutions to user problems (Ogu & Adekunle, 2013).

Traditional ESs were classified according to the used platform and method. Modern systems are much more universal. Today's expert systems have several logics for decision making, and not only "yes" or "no." It can be a fuzzy multi-criteria group decision-making model with a selection of rational criteria set to use in the weighted cost-effectiveness analysis (Zavadskas et al., 2017).

The Expert System Shell is responsible for managing and processing input and service requests from users and generating output; supporting the creation and modification of inference rules by knowledge engineers; translating the inference rules created into machine-readable forms; processing the information given by the user and the application layer modules, and relating such information to the concepts contained in the knowledge base through the inference rules; providing solutions for particular problems within the scope of its integrated knowledge domain; providing facilities for uncertain reasoning; providing facilities for knowledge representation (the knowledge representation knowledge) and editing the content of the knowledge base (knowledge base editor); and providing low level support to expert system components (such as retrieving metadata from and saving metadata to the knowledge base, building Abstract Syntax Trees during the

translation of inference rules, and other similar functions) (Ogu & Adekunle, 2013).

The User Interface receives requests from end-users and forwards them to the appropriate component of the ESS. For example, if the request received from an end-user asks for specific information from the expert system, it goes to the inference engine for its execution. Similarly, if the received request from the domain expert is meant for creating or updating inference rules, then the interface business logic dispatches the request to the knowledge base editor (S. Kumar & Prasad, 2015).

There are many different ES shells: Clips (Wygant, 1989), JESS (E. J. Friedman-Hill, 2007a), Exsys (Ibrahim et al., 2001), Vidwan (Mujawar & Jadhav, 2017), Knowledge Pro (S. Kumar & Prasad, 2015), K-Vision (Kholief et al., 2017), Age (S. Kumar & Prasad, 2015), Emycin (Venkata Subba Reddy, 2017), and XpertRule (Simiński & Xięski, 2016). The typical ES selection criteria include multiplatform support, support level, novelty, and price. The most popular ES shells are Clips, JESS, and Drools (Bobek, 2011; Xięski & Simiński, 2017).

The comparison of these ES shells is provided in Table 1.1.

Table 1.1. Comparison of expert system shells

Criteria	CLIPS	JESS	DROOLS
Novelty (solutions used, components, etc.)	-	+	+
Affordability/ Availability	+	+	+
Conclusion engine algorithm	RETE	RETE	RETE
Multiplatform	-	+	+
Knowledge visualization	-	-	+
Method	“Forward chaining”	“Forward/backward chaining”	“Forward chaining”
Modular rule grouping	+	+	+
Popularity	Medium	High	Medium

The criteria were selected through an expert survey. The experts were selected from the academic community in different countries. A group of eight experts was formed: four experts were from Vilnius Tech, one – from Austria, one

– from Vilnius University, one – from Belarus, and one – from Turkey. The experts were provided with a free-form survey asking them to indicate the most important criteria for expert system shells. After summarizing the survey, criteria mentioned by all experts were selected.

Based on the review performed, the JESS platform was selected considering the shell's high popularity and multiplatform criteria. Other JESS advantages also should be mentioned:

- Has all the same functions as Clips;
- Supports XML format;
- Compatible with Java classes and libraries;
- Supported JESS software with released updates. The latest JESS 8 version includes support for Android.

JESS is free for academic purposes but can be licensed for commercial use also. For the thesis experiments, JESS 6.1p8 and JESS 7.1p2 versions were used and JESS Software Research License Agreement were signed.

1.1.4. Rule Inference Algorithms

An expert systems shell is a system that facilitates the development of complete expert systems. They usually provide most of the important features: rule language, inference algorithm, explanation mechanism, user interface, and knowledge base editor.

The shell component in triangular form (Kumar & Prasad, 2015) is shown in Fig. 1.6.

The inference engine is like a “black box” that performs reasoning over the facts and rules. It receives end-user requests via the user interface and uses a pattern matcher to select the appropriate rule from the knowledge base. Now, the execution engine executes the rule in a certain order. There might be a possibility that more than one rule in the knowledge base matches the given fact. In such a situation, the execution engine falls into a dilemma, which to execute first (Griffin & Lewis, 1989; S. Kumar & Prasad, 2015). It uses different algorithms (like Rete Algorithm used in JESS) (Liu et al., 2010), by which the execution engine decides the order in which the rule is fired. This is called “Conflict Resolution.” Once rules are executed, an outcome provided by the inference engine is displayed on the user interface that the end-user is expecting (S. Kumar & Prasad, 2015).

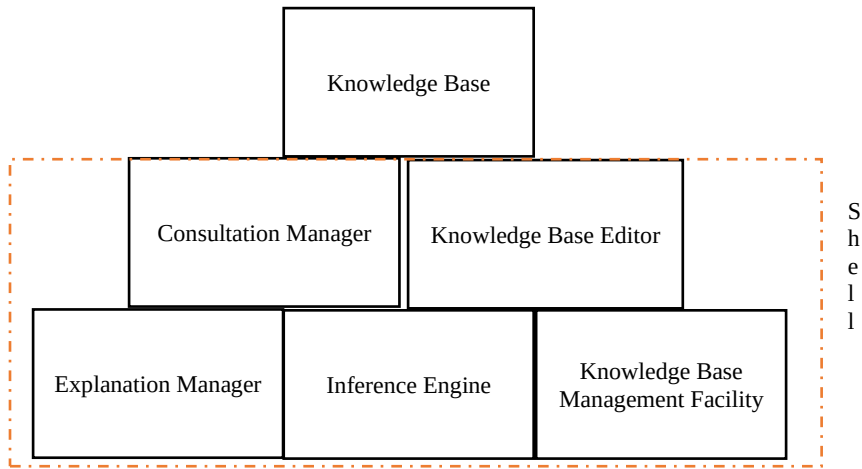


Fig. 1.6. Shell concepts for building an expert system

The main idea of the RETE algorithm (Liu et al., 2010) is that rules and facts change at different frequencies. More specifically, it has been observed that changes in the rules are relatively small. This has been used as a basis for making the operation of rule-based systems more efficient.

The RETE algorithm eliminates inefficiencies in model matching by memorizing the last recurring results of the cycle. Only new or deleted working memory items are activated using the model matching feature. Also, the RETE algorithm uses model matching only for those rules, for which groups could match those facts. In this way, the RETE algorithm enables the expert system to perform fast model matching.

The RETE algorithm is implemented by creating a network of interconnected nodes. Each node represents the left side (IF) of a rule or multiple rules. Entered or deleted facts are processed by the network of these nodes. There are incoming nodes at the top of the network and outgoing nodes at the bottom. Fig. 1.7 shows an example of the RETE network optimization (Hill, 2003).

The figure shows five input nodes, of which only three (x , y , and z) are different. Therefore, it is possible to adjust the network so that the two rules share three nodes. In the figure, each quadrilateral is a node. Small squares are single input nodes, large squares are two input nodes. Only the facts corresponding to the specified model come out of the node to the next node.

Nodes of two inputs memorize all the facts or groups of facts that come to one of the inputs and create groups of two or more facts for optimization in the output. The facts entered on the left and right of the node are segmented into separate memory blocks.

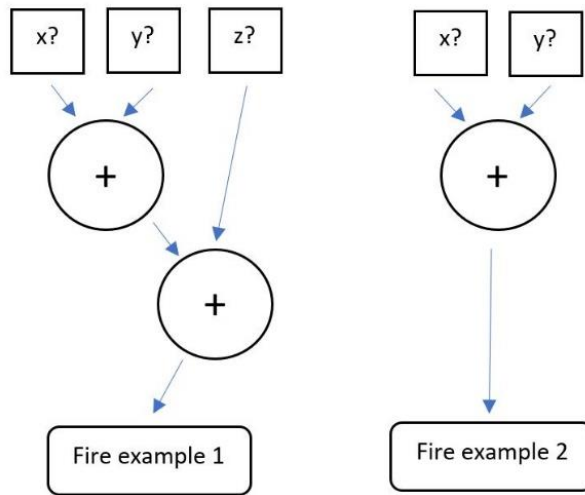


Fig. 1.7. Example of the RETE network optimization

The main idea of the TREAT algorithm is to exploit the conflict set support for temporarily redundant systems. The conflict set is explicitly retained across production system cycles, which allows for the following advancements comparing to Rete (Nalepa et al., 2011):

- 1) in the case of the addition of a working memory element, the conflict set remains the same, and the constrained search is performed for new instantiation of only those rules that contain the newly added working memory element.

- 2) a deletion from the working memory triggers a direct conflict set examination for rules to remove. No matching is required to process deletion since any instantiation of the rule containing the removed element is simply deleted.

Both RETE and TREAT are static networks that do not depend on system characteristics. This can create networks that are not optimal for certain knowledge bases. To solve this problem, the **Gator algorithm** was proposed (Berstel, 2002). It is based on Rete but additionally implements network structure optimization mechanisms based on specific knowledge base characteristics.

1.1.5. Review of Expert Systems Use for Risk Analysis

AudES is one of the first attempts to use ESs for information security management. AudES is a rule-based expert system. The purpose of this expert system was information security audit automation. AudES is designed to automate manual security audit procedures and reduce human input and errors that occur when a human operates a large amount of knowledge. It identifies problems but also gives recommendations for their elimination. In the final stage of the audit, the auditor

confirms the performed audit procedure, and the system archives everything. AudES is the intermediary between the auditor and the manual system security tool RACF (Resource Access Control Facility by IBM). This system focuses on detecting the factors identified in the system and gives recommendations (Tsudik & Summers, 1990).

The expert system has been developed using the commercial ESE IBM Expert Systems environment, selected for its successful application in diagnostics, planning, and decision-making. An important part of the system is the explanation module, which allows the user to ask questions about the actions performed by the system. All explanations and the path to conclusions are automatically recorded and saved in the system for later review and use, thus simplifying day-to-day tasks. Any knowledge or rules used by the system are taken from the RACF. Because the companies that use RACF are large, all security audit procedures in such companies are documented in detail. The authors say the system can audit multiple systems, thus avoiding unnoticed attempts to distribute each system distributively to a certain level as long as it does not exceed the systems response threshold.

The main disadvantage of this system is its relations with the RACF and the updating of the knowledge base, which is only possible through this system. Although the article mentions that AudES can be used in multiple systems, it fails to mention that all of them must be based on RACF by IBM. On the other hand, the system was developed at the IBM Science Center, focusing on the company's products. This is an advantage of the system for its users, as the entire knowledge base is taken from a common audit system, thus allowing to obtain up-to-date knowledge and transfer it easily to the ES (Tsudik & Summers, 1990). AudES asks for several questions with possible "yes" or "no" answers. Later, the flexibility of questions was increased, but the system was not supported, and the knowledge base was not renewed.

ESISA is an expert system for security audits (Kozhakhmet et al., 2012), which is based on ISO 27001 standard. It has the following parts of an expert system: the user interface, subsystem explanations, inference systems, the knowledge base, and database. Structural elements of the expert system perform the following functions (Bekezhanova & Atymtayeva, 2011):

- The user interface exists for direct interaction between ESISA and user, such as data entry, and displays the results of the processed data.
- The subsystem of explanations exists to clarify the actions of the expert systems logic for the users. The data from the subsystem of explanations are stored in the database table "Recommendation."
- The system of logical inference carries a substitution of values from the database in the field of parcel "If" (condition) of the rules knowledge

base, and in the case of filling in, all the parcels are ready to activate the processing rules, forming a conclusion in accordance with the part.

- The Knowledge Base is the main decision-making tool of the expert system, the main structure in which the knowledge of the expert system is stored.

Questions are separated by control objectives, and assessment is carried out in a complex of various procedures. ESISA is based on Fuzzy logic, which is a form of many-valued logic where the truth values of variables may be any real number between 0 and 1, inclusively. The value for the question is calculated as the sum of the coefficients of all entries.

These values, for example, the value of the negligence level and for the password strength, are subject to fuzzification into fuzzy subsets, like “weak,” “good,” or “strong” for a password’s strength and “low,” “moderate,” or “high” for negligence.

Fuzzy logic can be used in information security areas’ ES and in areas for analyzing the performance issues of a chemical process plant by utilizing uncertain, imprecise, and vague information (Panchal et al., 2019).

Some authors worked on improving ESISA and proposed to expand the knowledge base by adding more information security standards, i.e., ISO 27001, ITIL V3, and COBIT 4.1 (Bekezhanova & Atymtayeva, 2011). It is important to mention that it was a one-time knowledge base update where different information security documents were mapped, and no automated knowledge update method was proposed.

SISMS is a smart information security management system. It is on the list of risk management and risk assessment measures recommended by ENISA (ENISA, 2020). SISMS is a software package for information security management that uses an expert system to identify, implement, operate, monitor, review, maintain, and improve information security management. It is based on the international standard ISO 27001:2005. The system scans the organization’s network, including all hardware, and information (operating systems, specifications, operations and its efficiency, performance), scans open or filtered TCP and UDP ports and services and provides versions of open port services. In addition, it provides a network map of the organization based on the received network scan information (Desai, 2003). The system has three main functions: network monitoring, text-type add-on modules that set out other requirements and actions to be taken following ISO 27002:2005, and smart applications that determine the security status of an organization. All information is stored in a database used by the CLIPS-based ES to determine the organization’s current security status. The organization’s control requirements and rules to meet these requirements are based on ISO 27001 and ISO 27002.

The following are the main disadvantages of the software package: the system must be used by an employee with good IT knowledge; the system must be adapted to the customer, and the system can only be upgraded by the manufacturer according to the end-user's suggestions. It contradicts the information provided in the system's description that it covers a wide range of general information security management processes.

Both ESISA and SISMS require a good level of IT knowledge from the user and a lot of time for the knowledge base adoption for the specific enterprise. Currently, none of them is renewed despite the fact that standards were updated.

RAMeX is a prototype expert system for computer security risk analysis and management for small and medium commercial organizations. The RAMeX prototype was developed on the basis of previous research and the application of qualitative risk analysis to computer security in the commercial sector as a basis for the article (Bennett & Kailay, 1992). The aim of the RAMeX system is to perform a high-quality security assessment. RAMeX performs informal research quickly and is a cost-effective and practical management tool (Kailay & Jarratt, 1995). To avoid complexity, the terminology provided by ITSEC and DTI was used for the tool terminology. The tool was developed as a stand-alone system. Threats are classified into three groups: natural disaster, unintentional human error, and intentional human activity. The focus is on the last group. The system risk analysis and management methodology are developed from seven steps: asset identification, threat identification, impact identification, identification of existing security countermeasures, business impact assessment, assessment of needed security measures, and report generation.

RAMeX knowledge base structure consists of knowledge bases of assets, threats, vulnerabilities, and countermeasures. The work mentions that such a knowledge base structure facilitates the exchange of variables between the databases and the working memory.

So, RAMeX was the first known risk analysis-oriented expert system, and it was, in fact, a prototype for computer security risk analysis and management in SMEs. RAMeX could perform a qualitative risk analysis (only evaluating external threats) and provide a report with required countermeasures. This prototype of the system was not further developed.

IDES and NIDES are other expert systems mentioned in research papers, which are very specific (Lindqvist & Porras, 1999). They are dedicated to attack identification and don't give recommendations on problem solutions. So, in this thesis, they are not analyzed in more detail.

Wynyard Risk Management is a risk management platform with implemented information security management and policy management (Wynyard, 2015). A risk management system is based on the ISO 27001 standard.

It is not an expert system but rather an information security management tool that incorporates the main parts of the expert system.

The platform can be used to systematize the processes of risk identification, likelihood, and impact, thus simplifying the response to important processes, monitoring the progress of a particular process, and the risk mitigation process itself (Hicks, 2016). The system records all past and present incidents, lost data, etc., which must determine the results and conclusions provided by the system. Also, using the platform analytics module, various reports are possible. It was a commercial product developed by the Wynyard Group, but the development and support ended.

ESIA is an expert system that supports a systems auditor in risks evaluation and the choice of controls that reasonably protect the organization (Duque-Méndez et al., 2018). It is a typical expert system for systems security risk evaluation. The knowledge base represents the given facts and the actions of the systems audit expert under the methodology of a risks analysis. ESIA is a web-based system. From a technology point of view, it integrates a server and a web application with the logic server Prolog (Duque-Méndez et al., 2018).

When developing the system, the authors focused on extending the functionality of the system and, therefore, unlike other expert systems, integrated a Risk-Control matrix into the knowledge base. However, no further improvements were made compared to the classical expert system. The system has a total of 3600 rules described, and this is the biggest disadvantage of ESIA because the rules will become outdated and will have to be updated, which can only be done by the system developers.

The summary of expert systems related to the reviewed information security risk analysis is given in Table 1.2.

Table 1.2. Summary of expert systems related to the reviewed information security risk analysis

Expert system	Year	Notes
AudES	1990	It was one of the first attempts to use an expert system for information security management purposes. AudES is a rule-based expert system. Some commercial IBM products were used to develop this system, making it difficult to update the knowledge base. The system is no longer available.
ESISA	2011	It is an expert system for security audits based on the ISO 27001 standard and fuzzy logic. The expert system user must be IT-skilled. It is difficult to update the knowledge base. The system is no longer available.

End of Table 1.2

Expert system	Year	Notes
SISMS	2013	It is an information security management software package that uses an expert system to identify, implement, operate, monitor, review, maintain, and improve information security management. The system must be used by an employee with good IT knowledge; the system must be adapted to the customer, and the system can only be upgraded by the manufacturer himself according to the end-user's suggestions. The system is no longer available.
RAMeX	1992	It is a prototype expert system for computer security risk analysis and management for small and medium commercial organizations. It was one of the first attempts to use an expert system for information security risk analysis. This prototype of the system was not developed further.
IDES, NIDES	1999	Very specific expert systems are mentioned in research papers dedicated to attack identifications and giving no recommendations on problem solutions. So, in this work, they weren't analyzed in more detail.
Wynyard Risk Management	2015	It is a commercial risk management platform where information security management and policies management are implemented. Such system type can be called a Government, Risk, and Compliance (GRC) system. Its development and support ended, and the product is no longer available.
ESIA	2018	This is an expert system that supports a systems auditor in risks evaluation and the choice of controls that reasonably protect the organization. It is a typical expert system with an improved structure of the knowledge base but the still existing problem of the knowledge base updating.

In summary, such systems can automate just a part of the risk analysis process, they have their own knowledge base, and it is a complicated process to form and update it. Not one of the reviewed ESs is adapted for region-based security risk analysis. It is also important to mention that all reviewed ESs are based on the information security standards, where security controls are described, but there is a lack of information about regular updates of the likelihood, impact, and security controls.

1.2. Expert Systems' Knowledge Base Formation Methods

One of the biggest problems in using expert systems is to get the knowledge base updated (Sajja & Akerkar, 2010). This process is called knowledge engineering. Knowledge engineering is a branch of science that describes the development, maintenance, and improvement of knowledge-based systems (Gudas, Lopata, 2011). Knowledge engineering is a process of extracting the knowledge required for solving a problem from a human expert or another source, formally processing the acquired knowledge in a format understandable to the system and transferring it to the knowledge base. Such a process is called knowledge representation.

Knowledge could be obtained by interviewing domain experts and/or learning by experience. Several methods exist for extracting human knowledge. The knowledge base stores field knowledge, which is then used in the data mining process. The knowledge base can contain rules, templates, document specifications, or other structures. With their help, data mining queries are specified, and attributes and additional parameters are checked (Stravinskienė, Gudas, 2010).

The correct and complete recording of expert information in the knowledge base is one of the basic conditions for the correct functioning of the ES; therefore, a knowledge engineer creating an ES knowledge base must have at least a minimal understanding of the problem area to describe the rules correctly and logically (Janulevičius, Goranin, 2013). The reliability and efficiency of an expert system depend on the quality of the knowledge it has. The most important ES rule is that the knowledge base must be constantly developed and updated while the system is used.

One of the most difficult points in creating expert systems is creating a knowledge base. This part of the expert system stores system rules, patterns, facts, data specifications, and other structures (Elsharif & Abu-Naser, 2019). It allows the inference engine to give the right answer to the task (Dheir & Abu-Naser, 2019). There are several methods for knowledge base rule generation (Tandon et al., 2018):

Manual is the knowledge created by a real expert and knowledge engineer.

Automated is the knowledge created partly by humans and partly in an automated way.

Automatic is the knowledge created without the participation of human experts.

1.2.1. Review of Existing Manual Expert Systems' Knowledge Base Formation Methods

Interview. When researching methods for building an ES knowledge base, the interview appears to be the most common and commonly used method for knowledge acquisition. Interviews can be unstructured, structured, and other (Dipboye, 1994). Most knowledge engineer meetings begin with an informal interview to get to know the expert and gain an initial understanding of the structure of the problem area. A more structured interview is then used to assimilate knowledge.

The biggest advantage of using an interview is the natural process that is easily understood by the knowledge engineer and expert. However, an interview is more than just sitting down and talking to an expert. Regardless of whether this would be a structured or unstructured interview, it is often necessary to have more than one meeting, which makes the acquisition of knowledge slow and difficult. The interview is based on the expert's ability to clearly express and formulate the information used to perform the work. Unfortunately, it is often difficult for an expert to explain how they solve problems. Another factor is the ability and knowledge of the knowledge engineer: insufficient knowledge of the terminology and concepts in the problem area can make it difficult to control the process. Therefore, an interview is not always a reliable way to obtain detailed, objective, and well-organized descriptions of complex cognitive processes.

Questionnaire. According to Judith R. Olson and Henry H. Rueter, a questionnaire can be a very effective and efficient way to extract expert knowledge (Carter et al., 1992). It facilitates the assimilation of clearly defined knowledge. This method can be used to identify problem area objects and defined and indeterminate relationships between these objects. The source mentions that the main advantage of using questionnaires is that the knowledge engineer takes less time than the interview, it is an effective way of gathering information, and the expert can answer questions in their spare time. The main disadvantage of the questionnaire is that it cannot further explore the answers to unforeseen questions.

Task Observation. The essence of task monitoring is observing the work of an expert to determine how they make decisions by solving real problems in their field. Experts usually start their work from a problem. Using this method, the observer monitors the behavior and activities of the experts and can then ask questions. Task tracking is an easy way to gain knowledge, and how the interview is easy to understand for both the expert and the knowledge engineer (Jamshed, 2014). Monitoring actual performance can reveal a strategy for generating a conclusion and explain a misleading or incomplete problem-solving process. Task tracking is not suitable for achieving all knowledge acquisition goals and is a limitation of this method.

The biggest problem is access to experts and their jobs for monitoring. This issue may take a long time to resolve. Besides, an observed expert can behave unnaturally. Also, as a negative consequence, a large amount of information collected during the observation period may be present, but only a small amount of information may be useful in solving the problem.

Protocol Analysis is very similar to the task observation method, but the protocol analysis uses real or simulated problem-solving scenarios that the expert performs. Unlike the task observation method, the expert takes turns explaining their thinking processes as they solve the problem (Ericsson, 2008). The further in-depth analysis provides facts and rules that should be used in the knowledge base.

Using this method, both tacit and expressed expert knowledge can be gathered. The main advantage of using this method over the methods discussed earlier, such as the interview or task observation, is that the information gathered is directly related to the problem-solving process, and the knowledge engineer no longer must draw conclusions about problem-solving actions. Protocol analysis requires time to extract the appropriate expert knowledge for the model. Another disadvantage of the method is that it may not be able to absorb all the expert knowledge, and the action for solving the problem-solving process may affect the actual expert solution to a particular problem.

Interruption Analysis. Interruption analysis is another way to extract information from an expert (Olson & Rueter, 1987). As with the task monitoring method, the expert is monitored for how they solve current tasks (Burge, 2001). This method does not require the expert to take turns explaining their decision-making process, but when the expert performs some actions that the knowledge engineer does not understand, they interrupt the expert and ask for an explanation of the decision-making process.

The main advantage of the interruption analysis is that a knowledge engineer can capture the expert's knowledge at the moment when their focus is the greatest. The advantage of the interruption analysis is risky because interrupting an expert at an inconvenient time can lead to the analysis process not being repeated. This method provides the best result when used after the prototype system has been developed and compares the performance of the system with the performance of the expert.

Drawing Closed Curves is the specialized approach applied to all tasks in the problem area related to spatial connections between objects. The literature states that this is a direct method of knowledge acquisition. This method is not based on the expert's ability to orally explain the relationship between objects in the problem area. Instead, the expert is provided with several objects to be evaluated, and the expert is asked to indicate which objects are related. A closed curve is then

drawn around those objects. Closed curves are compared and evaluated for consistency.

Inferential Flow Analysis. According to J. R. Olson and H. H. Rueter, indirect flow analysis is a direct knowledge acquisition technique that allows obtaining clear expert knowledge (Olson & Rueter, 1987). Indirect flow analysis can be considered a separate form of an interview. This method uses domain-specific questions to determine cause-and-effect relationships between domain concepts. Interviews are used to create “causal networks” between different concepts. Applying this method is a simple and powerful tool for gathering information.

A review of only a small part of the methods for the non-automated formation of the knowledge base reveals fundamental problems. One of them is the time required for the process. Preparation is also an important step in the process. Meetings require many sessions, these sessions can take a long time, and there is a chance that the goal will not be missed during the meeting. Analysis of data collected during sessions can take days or weeks. Then the time taken to start building the ES becomes a serious obstacle in the process of building the system.

Other problems are related to selected experts. The chosen expert may have little work experience, or it may be a temporary substitute. Even if the expert’s knowledge is competent, they may perceive the process as a threat of losing their job if the processes are computerized. It may be that the expert is reluctant or uncomfortable disclosing their actual work methods.

Finally, problems can arise with the knowledge engineer. The methods discussed above are highly dependent on the skills and experience of the knowledge engineer. If the knowledge engineer does not delve deep enough into the problem area, does not understand the intended goals of the process, fails to establish contact with the expert, misrepresents the knowledge, etc., it can severely damage the whole process (Lethbridge et al., 2005).

On the one hand, a lot of data is accumulated, posing many challenges working with big data. Big data mining challenges relate to enormous and evolving data volumes, heterogeneous and autonomous sources, diverse representations, unstructured data, and required high-performance computing platforms (Wu et al., 2015)

Such shortcomings can turn the building of an ES into a very costly project. Therefore, there is a need to automate the development of the knowledge base for security risk analysis to minimize expenses. However, despite these shortcomings, it should be emphasized that the first intelligent systems were created by acquiring knowledge from experts in these ways precisely.

1.2.2. Review of Automated Expert Systems' Knowledge Base Rules Generation Methods

The analysis of expert systems has shown that rule-based expert systems are universal and, therefore, can be considered a proper solution for the task of risk assessment automation. Expert systems can be a proper solution for enterprises because they allow reducing costs for risk management (Batista et al., 2018). But to assess information security risks quickly and accurately, it is necessary to process a large amount of data about newly discovered vulnerabilities or threats to reflect regional and industry-specific information (Li et al., 2019).

The first attempts at system automation were designed to extend, maintain, or modify incorrect knowledge bases in existing systems. The interview system was integrated into the knowledge base in some way, with suggestions popping up for the expert, such as supplementing the knowledge to eliminate incorrect decision generation. The advantage of this method is that knowledge is entered directly into the system by the expert, there is no additional human mediator, i.e., the role of the knowledge engineer is reduced. The interview system uses the existing knowledge base to describe the correct solutions to the problem, so it is easier to understand the context and focus on the questions.

However, the dialogue generated by these systems was not modern, and questionnaire templates were prepared with domain concepts and limited possible answers. Typically, such an approach requires that the basic structure of the knowledge base and the underlying problem-solving strategies be static so that any changes in the knowledge can be easily adapted.

Protocol analysis process. There are several software tools that help the knowledge engineer analyze protocols, such as KEATS KA (1990), ACQUIST (1988), based on hypertext. The purpose of these software tools is to facilitate the analysis of the data collected by the knowledge engineer. This method allows the information gathered from interviews, protocol analysis, or textbooks to be divided into text. These fragments are assigned to concepts, and then the concepts can be formed into hierarchies. The knowledge engineer needs to define relationships to link concepts. The commercial PC-PACK tool has a workspace where entries can be annotated, edited, added, or highlighted with different colored markers.

Some protocol analysis procedures were more automated in KRITON software (1988). The record is automatically divided into sections separated by an expert, and semantic analysis is performed on each section to determine the text statement. The knowledge engineer then checks the suitability of the selected operators and arguments and tries to examine all the variables using the existing knowledge base of the problem area.

Text analysis methods. In the TOPKAT (1994) system, some methods for applying natural language processing methods and assimilating knowledge have

been applied. A lexical cursor is used to identify nouns during the transcription of a conversation with an expert in the problem area. Transcription – (Latin transcription – transcription): accurate recording of speech sounds according to pronunciation, in accordance with a certain system of signs and the rules of their use (Lithuanian dictionary). Those nouns that are repeated more frequently in a topic than in everyday speech are pre-defined as problem area concepts. The adjective immediately before the noun indicates a certain value of the concept attribute; the person is asked to provide a certain name for this attribute. This system performs a basic analysis that requires human intervention. Petter Stephen believes that the advantage of this method is less time required for knowledge analysis.

A very similar approach is applied in the KRITON (1988) and KITTEN (1988) systems for the acquisition of basic knowledge from textbooks in a problem area. It is assumed that words that are repeated more often in the text than others represent concepts of the problem area. These words were then used as a basis for gaining further knowledge from the expert.

1.2.3. Review of Automatic Expert Systems' Knowledge Base Rules Generation Methods

In a classical case, humans as real experts form expert systems' knowledge bases. However, it is a slow and expensive method causing one of the biggest problems for broader use of expert systems. Nevertheless, expert systems are used nowadays for solving actual problems, e.g., first-line IT service-desk, medicine, and law. Some researchers focus on this topic, and several methods of automating the ES knowledge base formation were proposed (Ivkin & Burlakov, 2020; Nourian et al., 2019; Wu et al., 2015).

The scheme of automated knowledge base formations methods is given in Fig. 1.8.

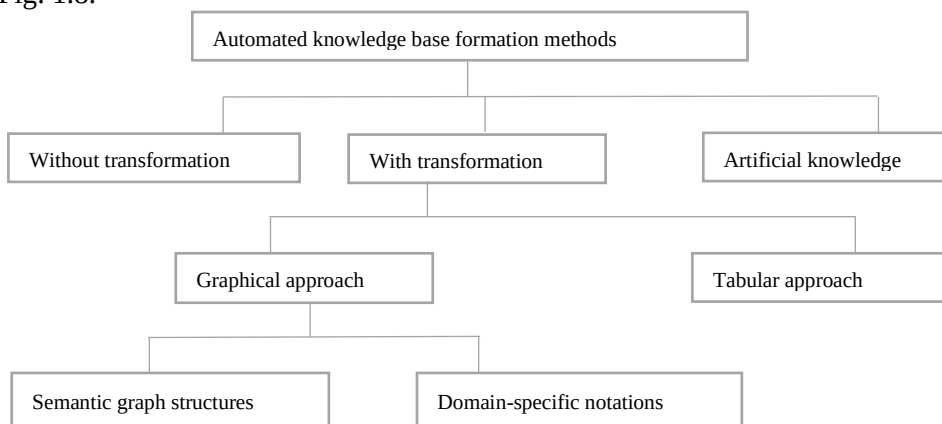


Fig. 1.8. Scheme of methods for the automated formation of a knowledge base

In general, all automated knowledge base formation methods can be classified into three approaches: (1) methods without additional transformations, (2) methods that require additional transformations, and (3) generation of artificial knowledge. More detailed descriptions are given in Table 1.3.

Table 1.3. Descriptions of automated knowledge base formation methods

Method	Description
Without transformation	This is a textual approach based on using existing knowledge sources directly in their native format, e.g., ontologies, and databases, without additional transformations (Feldman & Sanger, 2007). This approach is focused on programmers and is implemented in the form of a specialized editor. The main disadvantage of these methods is that it is difficult to update knowledge sources, or sometimes it is not possible at all.
With transformation	These methods are applied when the existing knowledge source transforms into ES knowledge base rules. There are several approaches based on data translation into program codes (Yurin & Dorodnykh, 2020):
Tabular approach	A tabular approach provides the construction of decision tables and their translation into program codes in knowledge representation languages. The standard formalism of decision tables and their specialization, e.g., eXtended Tabular Trees (Nalepa & Kluza, 2012), are used.
Graphical approach	A graphical approach provides the design of visual abstractions corresponding to the logical rule elements with their subsequent translation into program codes in knowledge representation languages. This approach is the most popular one and allows non-programmers to develop knowledge bases and expert systems. Within the graphical approach, the following areas can be identified (Yurin & Dorodnykh, 2020):
Semantic graph structures	This area is well represented by software for designing concept maps and ontologies.
Domain-specific notations	These notations provide modeling of logical and causal relationships.
Generation of artificial knowledge	Machine learning algorithms for knowledge-based formation are based on data classification (Bayes, Nearest neighbor method, regression, genetic algorithms). All of them are useful for prediction tasks where a human can't make the prediction obviously. The biggest disadvantage of such systems is that they are susceptible to errors. It can take a lot of time to discover the bug. Another limitation is related to the fact that the self-learning system needs a lot of high-quality data.

Methods without a needed transformation are cheaper but very static. Early studies on using existing knowledge sources directly in their native format considered that a problem did not exist until there was no need to change, update, add, or delete information (Akerkar & Sajja, 2009). The use of information sources directly in expert systems requires specific ES modification and lacks flexibility. Finally, it is very difficult or sometimes even impossible to merge several sources of information having different formats.

The second approach, when existing information sources are transformed into the ES knowledge base rules, is more superior and flexible, has more areas of application than the first, but requires additional methods of data transformation, additional time, and is sensitive to the correctness of data transformation (Sajja & Akerkar, 2010). Methods for the automated formation of expert systems' knowledge base using existing information sources are currently of great interest to researchers. Unfortunately, not too many methods were proposed for integrating IT security-related data into expert systems.

Another well-known method for knowledge base formation is machine learning. Ethem Alpaydin (2010) suggested a simple machine learning method for knowledge base formation (Kruse & Borgelt, 2003). There are several machine learning algorithms based on data classification (Bayes, nearest neighbor method, regression, genetic algorithms, etc.) (H. Xiao et al., 2017). All of them are useful for prediction tasks where a human cannot make the prediction obviously. Machine learning algorithms need no human interaction; they are self-learning systems and can work with various types of data. But the biggest disadvantage of such systems is that they are susceptible to errors (Mohamed, 2017). It can take a lot of time to discover the bug. Another limitation is related to the fact that a self-learning system needs a lot of high-quality data (Ristoski & Paulheim, 2016). It is not suitable for the information security analysis expert system since, e.g., the likelihood of a specific threat is a specific and dynamic value and, as mentioned above, there is a need for actual up-to-date data.

The analyzed knowledge acquisition methods are summarized in Table 1.4. The analysis of methods for the automated formation of a knowledge base for expert systems has shown that there isn't such a method that can be used for expert systems dedicated to information security risk analysis. There are some methods related to automated knowledge base formation or knowledge acquisition in general, but according to the identified disadvantages, they can be applied to other areas other than information security risk analysis.

The research on such methods has shown that early approaches to universal knowledge base formation automation were unsuccessful. Later, some specific approaches were proposed. The most popular were based on ontology integration and machine learning. The main disadvantage of ontology-based methods for risk analysis area is that data in an ontology is typically static, outdated, or suitable

only for one part of the risk assessment process (e.g., control selection), while machine learning is suitable when good datasets are available. Even though there are some automated methods for knowledge base formation, they are not suitable for creating an expert system's knowledge base for information security risk analysis responsible for analyzing threats in dynamic areas, such as malware and vulnerability threats.

Knowledge acquisition plays a critical role in constructing a knowledge-based system. It is the most time-consuming phase and has been recognized as the bottleneck of ES development (Chen & Rao, 2008). Therefore, there is a need to automate the security risk analysis process by introducing expert systems or decision support systems, which could help SMEs to perform an information security risk analysis without any special knowledge and without hiring security experts and making the security risk analysis process faster and cheaper (Powell, 1996), i.e., affordable for SMEs.

Table 1.4. Summary of knowledge acquisition methods

Method	Year	Notes	Authors
Early knowledge acquisition techniques	1986–2006	In the early years of developing expert systems, there was a prevalent idea that an expert and a knowledge engineer are the same person. But later, the research identified different roles and found that knowledge acquisition was the expert system's "bottleneck." After that, there were many related studies done and knowledge acquisition methods and techniques proposed, but not one was successful and remains useful nowadays.	(Babu, 2012; Boose, 1989; Cullen & Bryman, 1988; DeKeyser et al., 2001; Jasbir Singh Dhaliwal & Benbasat, 1990; Hoffman, 1987; J. Kim & Courtney, 1988; Wagner, 2006)
Matrix representation and mapping (MRM)	2008	This paper presents a matrix representation and mapping approach to facilitate the effectiveness of knowledge acquisition in building a KBS. The proposed MRM approach, which is based on matrix representation and mapping operations, comprises six consecutive steps for generating rules. The main disadvantage is that this approach doesn't form knowledge bases itself as it needs adoption for every source.	(Chen & Rao, 2008)

End of Table 1.4

Method	Year	Notes	Authors
MOKA methodology	2005	The paper focuses on knowledge acquisition and, specifically, on the use of the MOKA methodology to support knowledge acquisition for the development of knowledge-based engineering systems. MOKA methodology has three models: informal knowledge model, formal knowledge model based on Unified Modeling Language (UML), and neutral language knowledge model based on XML. This methodology has some limitations related to information sources and software tools, which weren't fixed.	(Preston et al., 2005)
A framework for artificial knowledge creation	2017	This paper presents a recursive framework for Artificial Knowledge Creation. This framework offers an analytical perspective on the constituent dimensions of artificial knowledge creation. This paper focuses on how the implementation of an AI agent in organizations is related to organizational knowledge creation. It starts by resuming the different kinds of machine learning used and introducing a taxonomy of Artificial Intelligence Learning Algorithms. Then, it resumes the five-stage framework for organizational artificial knowledge creation.	(Harfouche et al., 2018)
The acquisition of tacit knowledge based on an ontological model	2018	Tacit knowledge is one kind of knowledge, namely, personal knowledge, which cannot always be articulated in a coded form. It is implicit and appeals to the experience and know-how of the person who possesses it. In this paper, the authors propose an approach for representing and sharing tacit knowledge using ontologies. This approach is not based on transforming existing knowledge sources and cannot be used for the knowledge base's automated formation.	(Chergui et al., 2020)

Currently, sufficient information, including security information, such as standards, best practices, and security ontologies (Blanco et al., 2008), is collected and presented on the Internet in different forms (Gero & Maher, 2013; Sikos, 2019). To get a good quality knowledge base, it is important to get knowledge from multiple sources, e.g., combining extensive literature review and expert mining (Kapse et al., 2018).

1.3. Potential Sources for Expert System's Knowledge Base Formation Automation

Unified and formal knowledge models of the information security domain are fundamental requirements for supporting and enhancing existing risk management approaches (Fenz & Ekelhart, 2009). Some research (Karabey & Baykal, 2013; Lashkarara, 2017; Peltier, 2004; Shorabeh et al., 2019) has shown that the current information security management needs to be based on different kinds of information types.

As a fundament, static information is used, such as information security standards (e.g., ISO 27000 standard family) and best practice (German IT Grundschutz Manual, Cobit best practices family). On the other hand, the information security risk management field needs to be in tandem with the changing environment or, even better, in front of it all. The risk, by definition, is the likelihood that a threat will happen in the future but not in the past. So, information security risk assessment should include potential threats that can be actual in the near future. When we are talking about information security risk assessment, we should think about specific cases, too, not only general ones. It means that information security risk analysis is performed for specific information systems, and it is very important to have some knowledge about them: perimeter vulnerabilities, platform vulnerabilities, application vulnerabilities, and the possible ways to exploit them.

A literature review showed several possible sources for the formation of an expert system's knowledge base. The first and most traditional is real expert knowledge. Ontologies are used for expert systems also. There are some specialized knowledge generators for expert systems (Rios-Boutin et al., 2008; Speck-Planche & Kleandrova, 2020). Some researchers propose to use UML diagrams, Business Process Models (S. K. Kim et al., 2008; Nalepa & Kluza, 2012; Rhem, 2005). It is possible to use attack trees' data and data collected from the websites.

In summary, the information security risk assessment process needs to operate with different types of information according to the specific case. It can be static, standardized information for general information risk assessments. There is a need to operate a very dynamic risk to identify new threats. And there is a need to operate information from a very specific domain.

Based on the comparison (Table 1.5), three potential sources were selected for automatic knowledge base rules to generate an expert system for information security risk analysis. The same group of experts and methods was used to select the criteria as in the identification of the ES shell selection criteria.

Table 1.5. Comparison of data sources for knowledge base engineering (to use for information security risk analysis)

Source\Criteria	Dedicated to information security	Available specific information for risk assessments	Sufficient number of sources in the information security domain	Are given in textual format?	Suitability for automatic transformation
Real experts (Babu, 2012; Kireeva et al., 2019a)	+	+	-	-	-
Ontologies (Gao et al., 2013)	+	+	+	+	+
Knowledge generators (Yurin & Dorodnykh, 2020)	-	-	-	+	+
UML (Nalepa & Kluza, 2012)	+	-	-	+	+
MITRE (Strom et al., 2018)	+	+	-	+	+
Business Process Model (BPM) (Olifer et al., 2018)	-	+	-	-	+
Attack trees (Löhner, 2018)	+	+	+	+	+
WEB data (Malware descriptions, CVSS, NVD, CVE, CWE, CPE) (Ristoski & Paulheim, 2016)	+	+	+	+	+

The following five criteria were chosen for the selection of sources, which were mentioned by most experts during the free-form survey:

- Dedicated to information security means that the source has information in the field of information security.

- Available specific information for risk assessments means that the source has specific information, especially needed for risk analysis, such as likelihood, cost, impact, controls, or countermeasures.
- Sufficient number of sources in the information security domain means that there are enough sources that can be reused.
- Textual format means that the source can be available in textual format, which makes the transformation process easier.
- Suitability for automatic transformation means that the source is stored in standardized structures and formats.

The sufficiency of the number of sources was evaluated based on a combined evaluation by experts and the performed analysis. The sufficiency criteria for different sources vary due to the source nature.

All these sources can be used for an automatic expert system dedicated to information security risk analysis and knowledge base rules generation as there are sufficient sources dedicated to information security. Even more, they have specific information required for risk analysis, especially, such as likelihood, attack cost, impact, controls, and countermeasures. Quick research has shown that sufficient sources can be reused and are given in textual format, making the transformation process easier. And finally, all these three sources are suitable for automatic transformation. This means they are stored in standardized structures and formats. The general concept of automated data extraction is provided in Fig. 1.9.

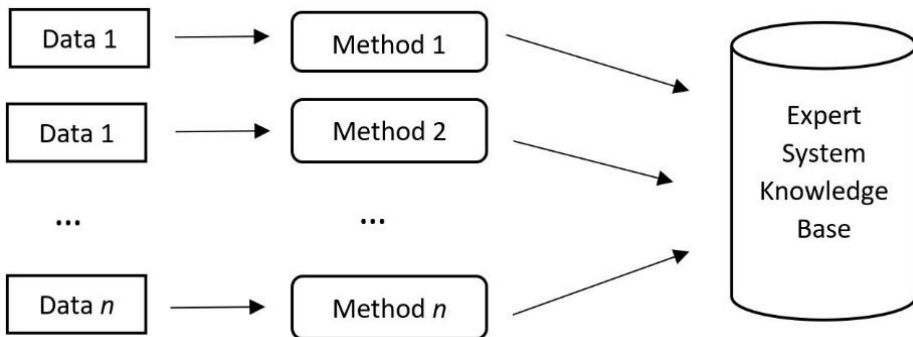


Fig. 1.9. General concept of automatic data extraction

In this work, the automation of the knowledge base of expert systems is based on the automatic transformation of existing sources into the rules of the knowledge base of the expert system, which supplement the existing or the newly developed knowledge base of the expert system. The final knowledge base is supplemented by rules obtained from the expert, if necessary. It should be noted that

the number of possible sources that may be sufficient may vary, so the number of proposed methods is not finite.

1.3.1. Ontology as a Source for Knowledge Base

The term ontology was introduced in informatics in 1990. The concepts of ontologies and semantic webs are relatively new, so they have no well-established definitions. In 1993, Gruber originally defined the notion of an ontology as an “explicit specification of a conceptualization” (T. R. Gruber, 1993). Ontologies are effective in solving the tasks of mapping and interconnecting different types of knowledge.

In a general case, the ontology of the subject area specifies the following elements: concepts, types of entities (objects and phenomena), hierarchies of concepts, interrelationships of entity types, valid rules, regularities about entity types and relationships, and case studies (Tankelevičienė, Sakalauskas, 2008).

Compared with a usual, not semantic database, an ontology has particular advantages that can be used for developing expert systems (Daraio et al., 2016). An ontology can be expressed in several ways, e.g., using graphical software, which makes ontology formation, maintenance, and use easier (Slimani, 2015). Therefore, an ontology presented in a particular software could be used for its automatic transformation into an expert system’s knowledge base. However, a deep understanding of ontology is needed since any change in it may require a change in the software’s source code (Kontopoulos et al., 2016). But these disadvantages can’t reproduce the expressiveness of ontology, and expert systems have their own inference engine, which is separate from the knowledge base.

The use of information collected in the form of ontologies for ES knowledge base formation seems useful for many experts. However, in practice, the current research focus on ontologies and expert systems is separated. The main idea of an ontology is to identify specific object classes and the relationships between them (Bova et al., 2014).

An ontology can be used in various business areas that can be divided into two parts: ontology as a vocabulary and ontology as content (Brank et al., 2005). The ontology used as a vocabulary allowed to develop the second generation of the web (T. Gruber, 2008). But ontology can be much more than a set of concepts or vocabulary as it has a lot of applications in artificial intelligence. In this context, an ontology can be used as an information source (Katz & Grau, 2005).

An ontology provides integrity and can be displayed by various graphical tools, has a common format, and can be apprehensible in other systems designed to work with ontologies (M\\as et al., 2005). For example, XP.K (eXtreme Programming of Knowledge-based systems) methodology is based on Agile principles: to start from the simplest model of an expert system and grow it with time

(Knublauch, 2002). A model should include only what is really needed to solve existing problems. A knowledge base is created from an ontology created by experts (Rick et al., 2010). The main problem of this approach is that it requires much time to develop the ontology from scratch on an iteration basis and does not make use of already existing collected knowledge, while it could be valuable to use already existing information security-related ontologies.

Some attempts (Otero-Cerdeira et al., 2015) were made to use ontology as a source of knowledge base formation, but the proposed approach destroys the idea of an ES that the knowledge base and inference engine should be separate. DAM-LJESSKB (Kopena & Regli, 2003) software was developed to transform DAML (DARPA Agent Markup Language) ontology to the JESS expert system's rules, but it can work only with a specific DAML ontology and a JESS expert system. Another sample is the DLEJena software that transforms an OWL 2 RL profile, compatible with pD semantic, into a Jena expert system platform (Meditskos & Bassiliades, 2008). The main disadvantage of these programs is that they transform only specific ontologies into a specific expert systems format.

On the other hand, several information security ontologies exist that can be used in the knowledge base formation of a risk-analysis ES. Currently, most of them are used for other purposes. For example, the ontology proposed by Sicilia et al. (2015) can be used as a tool to identify the level of system vulnerabilities according to the internal user's account configuration and system configuration (Obrst et al., 2014). The authors state that all illegal activity in the system is done by human resources, and internal users have more privileges than external. This tool is based on a taxonomy with user settings and includes different behavioral motivations, e.g., intentional and unintentional activity.

ROPE methodology and the related ontology are used for enterprise IT security evaluation and focus on business processes and risk management (Obrst et al., 2014). The ontology encapsulates well-known information security concepts, such as assets, vulnerabilities, threats, and controls.

There is a number of ontologies based on external security standards, such as ISO 27001/2, that can be used to align external standards with internal procedures (Fenz et al., 2016). Another security ontology is OntoSec (Obrst et al., 2014), which can be used by security engineers to set configuration effectively according to the existing security events.

According to Ramanauskaite et al. (2013), a new exhaustive ontology was proposed, which increased coverage of security standards compared to the existing ontologies and has better branching and depth properties for ontology visualization. It was used for the security standards mapping task and was linked with four security standards: ISO 27001, PCI DSS, ISSA 5173, and NISTIR 7621.

1.3.2. Web Data as a Source for Knowledge Base

Nowadays, information is digitized, and much of the latest information on cyber threats is primarily available on a variety of websites.

Web Data Mining is an important area of Data Mining that deals with the extraction of interesting knowledge from the World Wide Web. It can be classified into three different types, i.e., web content mining, web structure mining, and web usage mining.

Web Content Mining is relevant for the ES knowledge base formation. Since searching, comprehending, and using the semi-structured information stored on the Web poses a significant challenge because this data is more sophisticated and dynamic than the information stored by commercial database systems (Singh & Kumar Singh, 2010). This attribute becomes an advantage for the ES knowledge base formation as it allows to get especially dynamic knowledge from websites.

Web content mining uses different techniques to dig data. There are four techniques used by web content mining. Mostly, web contents data is in unstructured text form. For the extraction of unstructured data, web content mining requires text mining and data mining approaches (Malarvizhi & Saraswathi, 2013). Text documents are related to text mining, machine learning, and natural language. The main purpose of text mining is to extract previous information from the content source (Johnson & Kumar Gupta, 2012).

HyperText Markup Language (HTML) is a computerized markup language used to present content on the Internet. HTML primarily describes the logical structure of the pages of a message web: the titles, paragraphs, illustrations, tables, links to other documents, or other data of the documents and their constituent sections (Shannon, 2012). An example of HTML code is shown in Fig. 1.10.

```
<h2>Top - IN THE LAST WEEK</h2> == $0
▼<table id="country_stats_threats_list" cellpadding="0"
cellspacing="0">
  ▶<tr class="threats_link" data-link="https://threats.kaspersky.c
om/en/threat/HackTool.Win32.KMSAuto.gen">...</tr>
  ▶<tr class="threats_link" data-link="https://threats.kaspersky.c
om/en/threat/DangerousObject.Multi.Generic">...</tr>
  ▶<tr class="threats_link" data-link="https://threats.kaspersky.c
om/en/threat/HackTool.Win32.KMSAuto.ew">...</tr>
  ▶<tr class="threats_link" data-link="https://threats.kaspersky.c
om/en/threat/HackTool.MSIL.KMSAuto.di">...</tr>
```

Fig. 1.10. Example of a real-time map for Cyber-Threats in HTML code

The Web is typically the first source of information about new software vulnerabilities, exploits, and cyber-attacks. Information is found in semi-structured vulnerability databases as well as in texts of security bulletins, news reports, cybersecurity blogs, and Internet chat rooms (Joshi et al., 2013). Such information can be useful for information security risk analysis if the relevant information can be extracted from these sources.

One of the public and free resources of security information is the National Vulnerability Database (NVD)⁷ and its associated components, including Common Vulnerabilities and Exposures (CVE)⁸, Common Weakness Enumeration (CWE)⁹, Product Dictionary (CPE)¹⁰ and the Common Vulnerability Scoring System (CVSS)¹¹.

Today, various real-time attack maps, information about a malicious code, its prevalence, mode of operation, and other parameters are published on the Internet. Such publicly available information is shared by major antivirus vendors. The most well-known are Check Point Live Cyber Threat Map¹², Kaspersky Cyberthreat Real-Time Map¹³, FireEye Cyber Threat Map¹⁴, Bitdefender Cyber Threat Real-Time Map¹⁵, Fortinet Threat Map¹⁶, Imperva Cyber Threat Attack Map¹⁷.

Such information, especially information about malware, is relevant for expert systems dedicated to information security risk analysis. Any kind of cyber threat real-time map is very useful for such kinds of tasks because there is much updated information about malware, information is dynamic and relevant.

For this thesis, a real-time tasks implementation map for cyber threats with malware distribution was chosen as one of the sources for the expert system's automatic generation of knowledge base rules because it has specific information about malware prevalence by region, not available in NVD and related sources. Regional information about malware can be useful in information security risk analysis by regions. But it is important to mention that NVD and related sources, like CVSS, are also suitable for the formation of the expert system's knowledge base and can be planned for future work.

⁷ <https://nvd.nist.gov/>

⁸ <https://cve.mitre.org/>

⁹ <https://cwe.mitre.org/>

¹⁰ <https://nvd.nist.gov/products/cpe>

¹¹ <https://www.first.org/cvss/>

¹² <https://threatmap.checkpoint.com/>

¹³ <https://cybermap.kaspersky.com/>

¹⁴ <https://www.fireeye.com/cyber-map/threat-map.html>

¹⁵ <https://threatmap.bitdefender.com/>

¹⁶ <https://threatmap.fortiguard.com/>

¹⁷ <https://www.imperva.com/cyber-threat-attack-map/>

For the purposes of this thesis, one site was chosen to demonstrate the principle of web data transformation into rules of an expert system's knowledge base. Analysis has shown that Cyberthreat real-time map by Kaspersky Lab has the most detailed information about malware and also, like other providers, gives information in real-time and allows to filter malware distribution by region.

Malware on this site is grouped into eight categories listed in Table 1.6.

Table 1.6. Malware categories in Cyberthreat real-time map

Nr.	Category	Description
1.	OAS	On-Access scan indicates the detection of malware during traffic, when objects are accessed by opening, copying, executing, or saving operations.
2.	ODS	On-Demand Scan indicates malware detection traffic when a user manually selects the "Scan for viruses" option in the context menu.
3.	MAW	Mail Antivirus indicates malware detection traffic during a scan when e-mail new objects appear in the e-mail program.
4.	WAV	Web Antivirus indicates malware detection traffic during a scan when an HTML page of a website opens or files are downloaded.
5.	IDS	Intrusion Detection System indicates network attack detection traffic.
6.	VUL	Vulnerability Scan indicates vulnerability detection traffic.
7.	KAS	Anti-Spam, indicates suspicious and unwanted e-mail traffic.
8.	BAD	Botnet Activity Detection shows statistics about detected IP addresses of DDoS attacks and botnet Command & Control servers.

Such detailed information is relevant for information security analysis because it can allow to perform risk analysis more precisely and provide better recommendations. It is also important to mention that any mentioned Cyber Threat real-time map can be used for information security risk analysis, but in different levels of information details.

In summary, this is dynamic information, provided in real-time, grouped by region and type of malware that is needed for risk assessment.

1.3.3. Attack Trees as a Source of Risk Analysis Information

An attack tree (AT), first defined by Bruce Schneier as a “formal, methodical way of describing the security of systems based on varying attacks,” is typically a graphical representation of vulnerabilities within an IT system, although it can be expressed in a text format which is not graphical in nature (Moore et al., 2001; Schneier, 2015). Whether graphical or text format is represented, there are some similarities between these two representations.

Every path through the attack tree represents a unique attack vector through the organization and shows what vulnerabilities an attacker can leverage to obtain access to sensitive material (Moore et al., 2001). At every level or a leaf of the attack tree, there are different vulnerabilities, some of which must be combined to cause a breach and some which can cause a breach by themselves. These are known, respectively, as “and-” and “or-” decompositions and are represented differently based on whether the graphical, text, or combination format of the attack tree is used (Moore et al., 2001).

It is up to the creator of the attack tree as to how much information is shown and in what formats the attributes are represented, as no formal standards exist (Huistra, 2016; Kordy et al., 2014; Saini et al., 2008; Schneier, 2015). Despite the lack of formal standards, there are some recommendations for attack tree generation, such as creating the trees, measuring the probabilities of each leaf, removing or marking improbable attack paths, generating countermeasures, and applying the most appropriate countermeasures to leaves (Salter et al., 1998; Shostack, 2014). An example of an attack tree, based on an attack description provided by Kordy et al. (2010), is shown in Fig. 1.11.

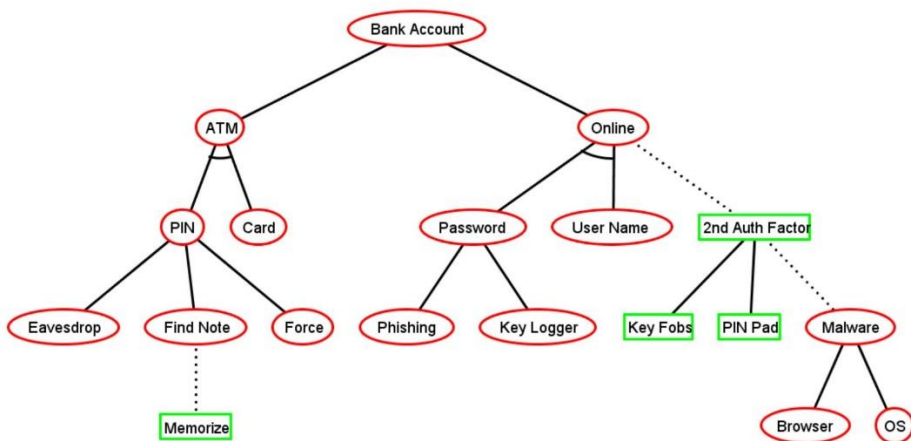


Fig. 1.11. Example of an attack tree’s graphical format (an attack on a bank account)

Each independent level is called a “leaf” node, and vulnerabilities are progressed during an attack from the bottom to the top, which is called the “root” node (Schneier, 2015). For this attack tree, straight horizontal lines represent “and-” decompositions while curved horizontal lines represent “or-” decompositions.

There are several programs that can be used to aid with the graphical representation portion of the attack tree creation process.

However, there is still a large amount of input required on the part of an expert person with manual attack tree generation (Kordy et al., 2013; Sheyner & Wing, 2004). Many of these graphing tools also include analysis functions to ensure that data is entered in an accepted format and to analyze the paths from leaf nodes to the root node. Text formats are often done when dealing with more complex attack trees as they are easier to read than a complex and branching graphical format with dozens of paths (Schneier, 2015).

The easiest way of working with attack trees for a human expert is the graphical format. However, for integration into information systems, the text format is preferred. For solving this problem, researchers (Gadyatskaya, Jhavar, et al., 2016; Vigo et al., 2014; Wang et al., 2012) have proposed methods for transforming attack trees from the graphical format to text and vice versa. This has increased the usability of attack trees. Nowadays, attack trees are widely used in the fields of defense, risks analysis, threat modeling against information, electronic systems, computer control, or physical systems. Based on the analysis of attack trees’ data, practitioners can define actions to reduce or annihilate risks. A significant barrier in integrating attack trees into the knowledge base of ES is that attack trees can become largely complex and, thus, hard to specify (Pinchinat et al., 2015).

The eXtensible Markup Language, or XML, is a language created by the World Wide Web Consortium (W3C) to store and transport data in a format which would be readable to computers and humans (W3Schools, n.d.). It is designed to be extremely flexible and allows creators to define their naming conventions within the XML structure. For example, the attack tree in Fig. 1.12 is an example tree from the ADTree attack tree graphing tool named “RFID Communication Block,” which was exported to XML by the tool.

This variability in XML formatting raises problems within the attack tree conversion process as each tool may define attributes using different names. For example, in line 11 of the figure text, the term “switchRole” indicates that the node “Secure Warehouse” is a countermeasure node rather than a vulnerability node.

In the universal metamodel for the attack trees format of this same file, “switchRole” is only called “role,” but this means any program that wants to extract and work with this XML data must be programmed to find all names for these attributes.

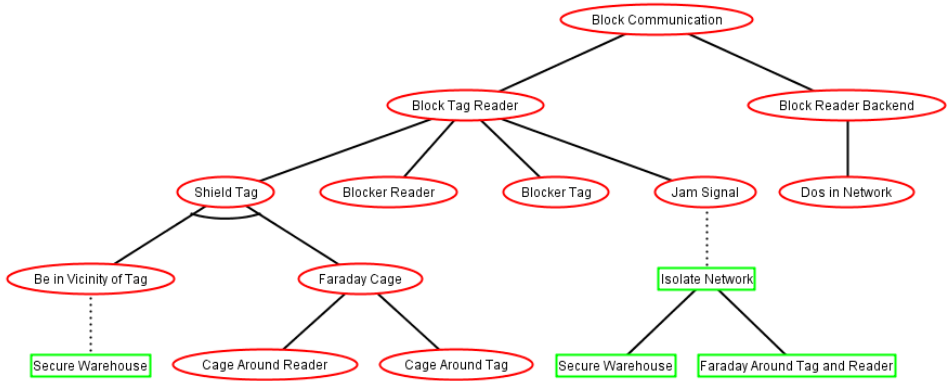


Fig. 1.12. RFID Communication Block Attack Tree

The result of this export is presented in Fig. 1.13.

```

<?xml version='1.0'?>
<adtree>
  <node refinement="disjunctive">
    <label>Block Communication</label>
    <node refinement="disjunctive">
      <label>Block Tag Reader</label>
      <node refinement="conjunctive">
        <label>Shield Tag</label>
        <node refinement="disjunctive">
          <label>Be in Vicinity of Tag</label>
          <node refinement="disjunctive" switchRole="yes">
            <label>Secure Warehouse</label>
          </node>
        </node>
      </node>
      <node refinement="disjunctive">
        <label>Faraday Cage</label>
        <node refinement="disjunctive">
          <label>Cage Around Reader</label>
        </node>
        <node refinement="disjunctive">
          <label>Cage Around Tag</label>
        </node>
      </node>
    </node>
    <label>Jam Signal</label>
    <node refinement="disjunctive">
      <label>Isolate Network</label>
      <node refinement="disjunctive">
        <label>Secure Warehouse</label>
      </node>
      <node refinement="disjunctive">
        <label>Faraday Around Tag and Reader</label>
      </node>
    </node>
  </node>
  <node refinement="disjunctive">
    <label>Block Reader Backend</label>
    <node refinement="disjunctive">
      <label>Dos in Network</label>
    </node>
  </node>
</adtree>

```

Fig. 1.13. RFID Communication Block Attack Tree in XML

Some tools can be used to aid with the graphical representation portion of the attack tree creation process, although there is still a large amount of input required

on the part of an expert person with manual attack tree generation (Kordy et al., 2013; Sheyner & Wing, 2004). While some researchers have tried to overcome this via crowdsourcing, others have attempted to utilize AI to generate the attack trees while an expert person reviews the automated creation results (Audinot et al., 2017; Jhawar et al., 2018; Tentilucci et al., 2015).

One common task which requires the use of an expert person is verification that the vulnerabilities expressed by an attack tree correlate to the system for which the attack tree is designed (Audinot et al., 2017). Research has been performed with the goal of reducing the expert-required workload, including attempts at utilizing vulnerabilities databases, such as the National Vulnerability Database (NVD) and Mitre Organization's Common Attack Pattern Enumeration and Classification (CAPEC) (Jhawar et al., 2018). While this research has shown considerable promise, there remains a need for experts at multiple steps in the attack tree generation, verification, and review stages. By inputting these attack trees into an expert system that has been configured to present an expert opinion, the need for an expert person could be further minimized.

Several methods for using attack trees for IT security risk assessments were proposed. One of such methods is IT security risk assessment for a ship control system based on attack trees (Shang et al., 2019). This method uses attack trees directly, not transforming to ES knowledge base rules, and is dedicated to supervisory control and data acquisition systems. Another example is the method to find the optimal set of countermeasures (Fila & Widel, 2020). This method is a framework relying on attack-defense trees and oriented into modeling in The Attack-Defense Tree Tool 2.0. The main limitation of such methods is that they use attack trees directly as sources and are not as universal as needed for expert systems.

1.4. Related Research in Lithuania

The topic of expert systems research in Lithuania is also relevant. This topic is not new in Lithuania, and research is still being carried out actively.

One of the first efforts dealing with the problems of artificial intelligence and expert systems was published in 1996 (Kulvietis et al., 1996). Broniukaitis (Broniukaitis, 1997) said that ESs solve tasks that would not otherwise be possible without the help of an expert, and this can be repeated without the participation of an expert. Kaklauskas (Kaklauskas & Zavadskas, 2010) added that nowadays, expert systems generally help to free a professional from solving some difficult but clearly articulated tasks.

The problem of updating the knowledge base has not remained untouched in solving the problems of expert systems' application. Kaminskas (Kaminskas

et al., 2005) created an expert system that allows the user to extend a database, change information, and effectively collaborate in creating, using, and controlling information with other welding specialists. Stravinskienė (Stravinskienė & Gudas, 2010) analyzed the application of data mining in enterprises and concluded that it is applicable not only to large but also to medium-sized enterprises.

To solve known problems of the ES knowledge base formation, it was proposed to use domain ontologies. Some authors present an approach for the development of an intelligent advice system for fault diagnostics based on domain ontology for supporting the identification of BIOS-related computer problems (Tankelevičienė & Sakalauskas, 2008).

Later, ontologies were actively used in various fields. Misiulienė proposed an approach to the multiplex use of ontology in e-learning (Misiulienė, 2012). She developed a metamodel for presenting educational material from ontologies.

Some other authors proposed methods of ontology-based transformations. Damaševičius and others proposed to represent domain variability using feature models enriched with lightweight domain ontology (Damaševičius et al., 2008). Authors transformed such models into generative component specifications using meta-programming techniques.

Not only an ontology allows reusing knowledge or making assumptions within a domain, but it also enables the analysis of every piece of knowledge. It was observed and used in digital forensics, where an ontology-based transformation model and a framework to develop an ontology-based transformation system in the digital forensics domain were proposed (Grigaliunas et al., 2017).

An ontology can be used for security standards mapping. Security ontology can be used to simplify the mapping of more than two security standards. This solution suggests mapping all security standards under one security ontology (Ramanauskaite et al., 2013). These authors, in their research, mapped four security standards (ISO 27001, PCI DSS, ISSA 5173, and NISTIR 7621). This and other ontologies can be used to form an expert system knowledge base of security controls. Security standards mapping and defining the minimum security baseline topic were developed in later research where other techniques, like graphs, were used (Olifer et al., 2019).

In summary, expert systems and other types of artificial intelligence are a perspective topic of research. Many efforts are contributed by authors from different areas of artificial intelligence application (Damaševičius et al., 2021; Maskeliunas et al., 2019; Venčkauskas et al., 2015) and decision making (Grigaliunas et al., 2020).

1.5. Conclusions of Chapter 1 and the Formulation of Tasks

1. The literature review has shown that information security risk analysis is crucial, and expert systems can help this assessment to be made by a specialist without information security expertise, but the main problem of expert systems is timely and sufficient updating of the knowledge base, which is a long and expensive process. On the other hand, based on scientific publications, it has been established that there are enough various sources of information security knowledge that are suitable for the information security risk assessment process; therefore, the automatic transformation of such knowledge into expert systems knowledge base is relevant and necessary.
2. It can be concluded that automated generation and updating of the expert systems' knowledge base for information security analysis is a relevant topic and can be solved by automatically transforming freely available information security sources. The analysis has shown that to date, no method has been proposed that would automatically generate a knowledge base of expert systems using freely available data from different sources. Non-automated or automated methods require the involvement of experts, and existing automatic methods have disadvantages. Re-using information without transforming it into ES rules is a cheap but static method and is, therefore, unsuitable in an ever-changing environment, such as information security. Existing universal information transformation methods can be applied in some other fields but have proved ineffective in the field of information security as they failed to transform various specific information in this field, such as threats, likelihood, complexity, and controls.
3. After the analysis of expert system platforms, according to the defined criteria, the JESS expert systems shell was selected, which is suitable and sufficient to implement the tasks set in this work. JESS was chosen because it is a supported and updated system, free of charge for scientific purposes, retains the advantages of other shells, uses the widely used language of the CLIPS rules, supports the XML format, and is widely used by other authors.
4. After the analysis of the use of possible information sources for the generation of the expert system's knowledge base for the analysis of information security risks, several sources were determined that are freely available and have different necessary information for risk analysis. From the sources analyzed, three different sources were selected according to the set criteria, such as suitability for information security, adaptability to

risk assessment, a sufficient number of information sources, availability in the text format, and suitability for automatic transformation: an ontology, web data, and attack trees, which can complement the expert system knowledge base with different types of information and ensure sufficient content of the knowledge base of the enough-scale information security risk analysis system of the expert system. Selecting a variety of information security sources with different refresh rates, levels of data detail, amount of information available, and other criteria and combining such sources into a single knowledge base would, in an automated way, create and update the expert system's knowledge base to perform information security risk assessment using an expert system without being an information security specialist.

Based on the conclusions, the following tasks are formulated to achieve the goal:

1. To propose an automated knowledge base generation method for expert systems for information security risk analysis, based on the transformation of information security ontologies into expert system's knowledge base rules and complementing the knowledge base with descriptions of information security controls.
2. To propose an automated knowledge base generation method for expert systems for information security risk analysis based on the transformation of web data containing malicious code descriptions into expert system's knowledge base rules and complement the knowledge base with dynamic information on the prevalence of a malicious code by regions.
3. To propose an automated knowledge base generation method for expert systems for information security risk analysis based on the transformation of attack trees from various sources into expert system's knowledge base rules that include information relevant to information security risk assessment, such as threat likelihood, cost, complexity, threat exploitation time, and countermeasures.
4. To develop a prototype of an expert system for information security risk analysis in the JESS platform, where the knowledge base would be updated by the methods proposed in this work with automatically generated rules, and to perform experiments that would allow to validate the proposed methods, evaluate the obtained results, and define the objectives for future work.

Automatic Approach of Ontology Transformation into Expert System's Knowledge Base

This chapter proposes an approach that allows automatic transformation of existing information security ontologies into rules of an expert systems' knowledge base. Unlike other methods, it does not integrate ontology directly but utilizes the transformation of ontology elements into rules of the expert systems' knowledge base: OWL RL subset of ontology is segregated into RDF triplets, which are transformed into the Rule Interchange Format (RIF); RIF rules are converted into JESS knowledge bases rules.

The proposed method and results were published in an international journal (Vitkus et al., 2019).

2.1. Ontology Integration via Rule Interchange Format

This chapter presents an approach that allows the automatic transformation of existing information security ontologies into a ruleset of an expert system's knowledge base grounded in the transformation of ontologies into RIF, and later,

using existing tools, transforming it into a JESS-based expert system. The main disadvantage of existing methods is that they transform only specific ontologies to only specific expert systems. A method that uses the intermediate RIF format solves some of these problems because it makes the process of ontology transformation into rules of an expert system's knowledge more universal and widely applied.

2.1.1. Rule Interchange Format

In mid-2010, the W3C consortium issued a new recommendation, the Rule Interchange Format (RIF), for systems to exchange rules. Although RIF was developed for other purposes, its application would allow efficient and accurate transfer of rules from ontologies to expert systems.

The development of the RIF standard started in 2005. According to the official W3C documentation, the idea of developing this standard was “intimidating” (Klyne, G. & Carroll, J., 2004). The RIF group took four years to develop it, and there are still many places to improve it. The development of this standard was a difficult task for three main reasons:

- Different understandings of what a rule is. There are many understandings of rules, such as deductive rules, normative rules, production rules, reactive rules, etc.
- Even in a general category of rules, systems are different both semantically and syntactically.
- Existing Semantic Web standards (such as RDF and OWL) need to be successfully integrated into RIFs, and practice and analysis show that their ability to represent rules requires detailed analysis.

Unlike other Semantic Web standards (RDF, OWL, SPARQL), RIF was not developed to become the standard language for describing rules but the standard for transferring rules from one system to another. This has become relevant because of the many different standards for writing rules: some systems even have their own specific syntax that is not used anywhere else. Regardless of the original objectives, RIF can be used like any other rule language, thus avoiding the conversion of rules from RIF to the desired system.

To avoid the same mistakes they made when creating the ontology language standard; the developers left it possible for motivated experts to add standard dialects with new functionality. Although such extensions would not be standard, they could become w3c standards if successfully applied.

2.1.2. Web Ontology Language Integration with Rule Interchange Format

A method is proposed to avoid spending time resources developing knowledge bases from scratch and aiming to solve the limitations specific to earlier research in ontology transformation to knowledge base rulesets. The method allows transforming existing information security ontologies into an expert system ruleset via the universal RIF format that is later converted into the format of a specific expert system. In fact, the method can be applied for any ontology-type transformation into any ES knowledge base with some additional modifications related to the syntax of ES rule-defining language.

RIF has two dialects: based on logic and based on rules. Dialects based on logic include logical languages, such as Horn logic and others. Dialects based on rules include productive rules. The same rules are used in expert systems (Group & June, 2010). RIF PRD (Production Rule Dialect) is a dialect based on production rules. Production rules semantic have the format “IF condition THEN activity.” Fig. 2.1 shows an example of RIF PRD.

```
Document (
  Prefix(rdfs <http://www.w3.org/2000/01/rdf-schema#>)
  Prefix(imdbrelf
    <http://example.com/fauximdbrelations#>)
  Prefix(dbpediaf
    <http://example.com/fauxibdbrelations#>)
  Prefix(ibdbrelf
    <http://example.com/fauxibdbrelations#>)

  Group (
    Forall ?Actor (
      If Or(Exists ?Film
        (imdbrelf:winAward(?Actor ?Film))
        Exists ?Play
        (ibdbrelf:winAward(?Actor ?Play)) )
      Then assert(dbpediaf:awardWinner(?Actor))
    )
    imdbrelf:winAward(RobertoBenigni LifeIsBeautiful)
  )
)
```

Fig. 2.1. Example of RIF PRD

Like other rule languages, RIFs can be written in a variety of formats (XML, Turtle, and others), but the XML format for data exchange is recommended, so only XML syntax will be used in this work.

2.1.3. Transforming Web Ontology Language (OWL 2 RL) to Rule Interchange Format (RIF CORE)

RIF was created in a way compatible with other W3C standards. That means it can be compatible with RDF and OWL ontology languages. Transferring these ontology languages into RIF can be done not only in the form of rulesets but also in RDF triples (subject, predicate, object) and OWL axioms with RIF rules, e.g., RDF data: S(Subject), P(Predicate) and O(Object) are transferred to the RIF format in a form: [(P->O)].

For example:

ex:Peter ex:isBrother ex:John;

ex:John ex:isFather ex:Paul;

states that Peter is the brother of John and John is the father of Paul.

In the *RIF* transferring to:

ex:Peter [ex:isBrother -> ex:John];

ex:John [ex:isFather -> ex:Paul];

RIF rule that uncle is the brother of father:

Forall ?x ?y ?z (?x[ex:isUncle -> ?z] :- And(?x[ex:isBrother -> ?y] ?y[ex:isFather -> ?z]))

states, that if *x* is the brother of *y* and *y* is the father of *z*, then *x* is the uncle of *z*.

Production rules are rules used by rule-based expert systems. As mentioned in the previous chapters, they are expressed as follows:

IF ... THEN ...

RIF-PRD allows you to transform rules from one expert system to the RIF format and then from it to another expert system.

The rules written in RIF look like this:

FOR <variables> WITH <matches> ,

IF <condition> THEN <action> .

Or

FORALL variables (actions: - AND (matches, conditions)

An example taken from Yu (2011):

Forall? Customer? PurchasesYTD (

If And (? Customer # ex1: Customer

? customer [ex1: purchasesYTD>? purchasesYTD]

External (pred: numeric-greater-than (? PurchasesYTD 5000)))

Then ex1: Gold (? Customer))

This thesis used the OWL RL profile as a source. OWL RL rules can be divided into four categories (which may overlap):

- Triple structure rules: RDF triplets. Transformation to RIF is trivial.

- Listing rules: RDF lists. Transforming in two ways – transforming into a recursive set of rules or transforming into triple structure rules.
- Inconsistent rules: RDF graph inconsistencies, which are expressed in first-order rules. There are several ways how to perform the transformation.
- Data type rules: type's comparison and verification. OWL and RIF-supported types are transformed directly.

Accordingly, OWL RL can be transformed into RIF rules. However, some limitations still exist. These limitations were introduced by OWL founders to provide maximum flexibility without sacrificing the reliability of the calculations. So, it can be said that the violation of at least one limitation affects the results of an expert system.

2.2. Method for Transforming an Ontology into Rules

The transformation process consists of two parts: transformations of fixed rules and ontology rules obtained from its T-Box axioms. In the OWL 2 RL profile, it is possible to write not only ontology axioms but also additional logical rules. However, this thesis uses OWL 2 RL as a subset of ontology axioms suitable for expert systems, and the aim is to transform classes and relationships between ontologies.

Ontologies can be incredibly complex structures, and at first glance, the conversion of each element is a complex process that requires reviewing all the relationships of that element to other elements. Instead, the proposed method in this work initially scans the ontology and saves it as a set of RDF triplets. No matter how complex an ontology is, it can always be written in a set of RDF triplets. The only problem is the correctness of the ontology, which the method of conversion cannot verify. This can be done with existing tools, or it can be trusted that the imported ontology is compiled correctly. Collected RDF triplets can be analyzed individually or in groups if several of them are inseparable.

In the next step, the RDF triplets are checked according to the conversion table, according to the convertible RDF triplet patterns. RDF triplets can be various, but their essential part, the predicate, can take values from the exhaustive list (due to the property of the ontology formality). The remaining two triplet elements, the subject and the object are specific concepts or attributes described in the ontology that fill the transformation template (there are several exceptional cases where the template itself changes according to them, e.g., *owl: maxCardinality*).

2.3. Method Implementation for Transforming Ontologies into Rules of an Expert System's Knowledge Base

The method presented in the previous chapter was implemented using C# language, and the general view is given in Fig. 2.2.

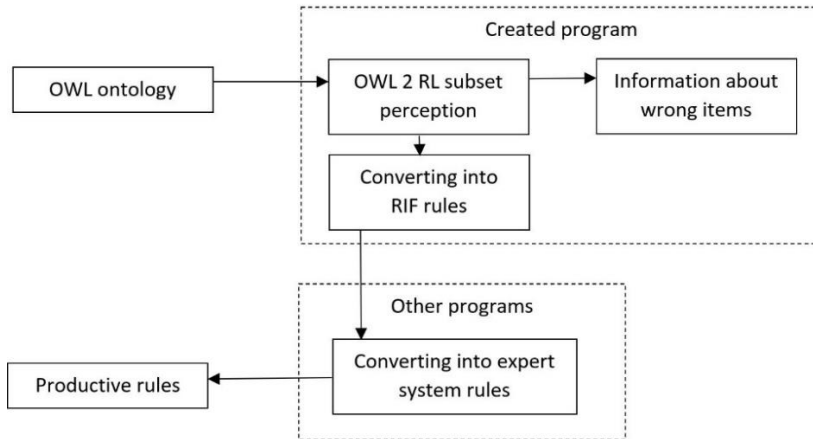


Fig. 2.2. General view of the proposed method

The main steps of the method are presented in the figure:

1. During the first step, the developed tool converts the information security ontology into RIF standard.
2. During the second step (RIF transforming into expert system production rules), the RIF PRD profile was used, already included in the RIF Core profile.

The obtained RIF rules can later be transformed into the syntax supported by the ES or used directly if ES supports the RIF format, like XSB, JESS, and ASP.

2.3.1. Design of an Ontology Transformation System

The system has five main parts: (1) user interface, (2) ontology scanning and (3) analysis, (4) transformation of the ontology into the form of RDF triplets (subject, predicate, object), and (5) transformation into RIF rules.

The detailed method activity diagram is shown in Fig. 2.3. The yellow border depicts actions proposed and implemented in this thesis, while actions outside the yellow border are performed with the help of already existing methods and tools.

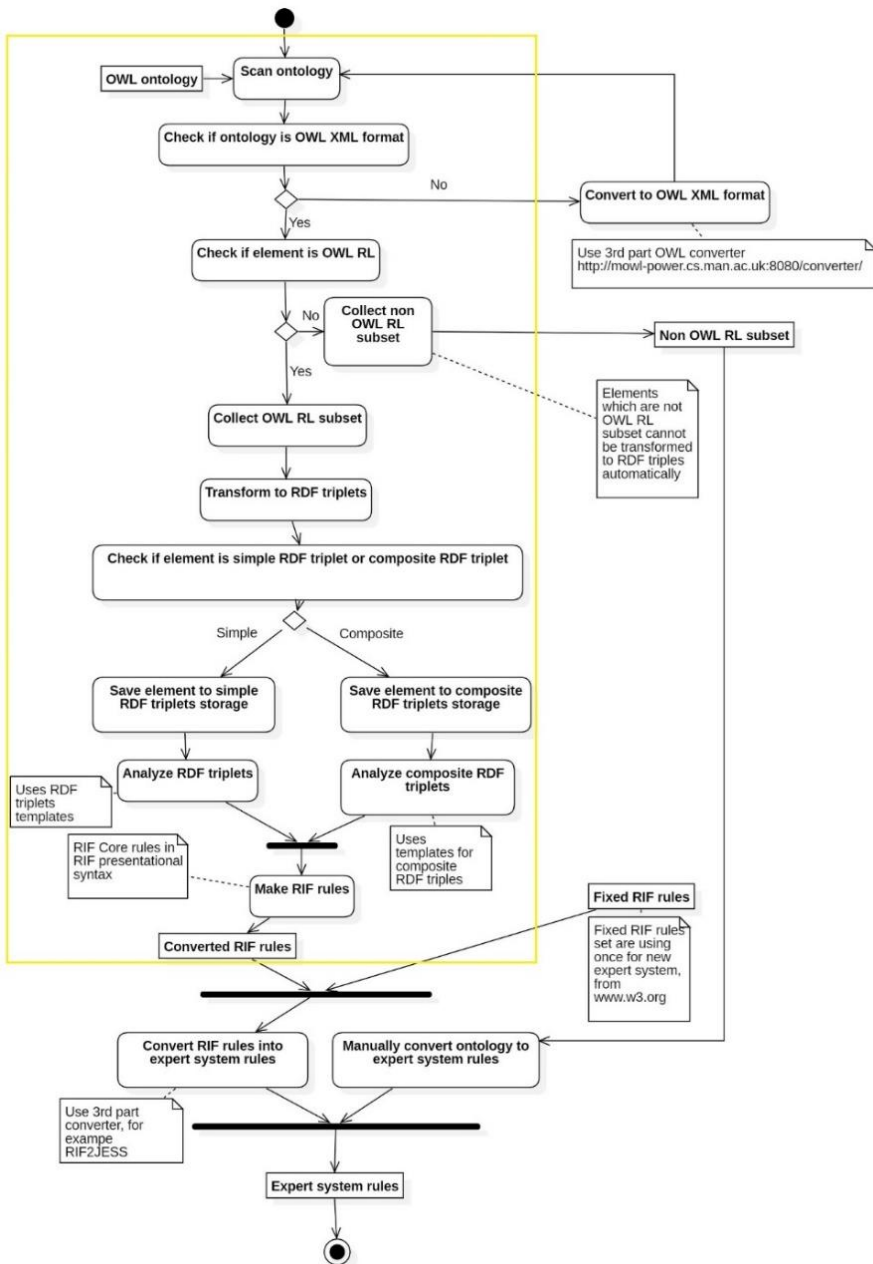


Fig. 2.3. Detailed method activity diagram

The model consists of primary data – ontology (OWL). The transformation of RIF rules is performed in several steps:

- Scan ontology. OWL/XML ontology is scanned and checked if the ontology is in the OWL/XML format since the developed tool supports only the most popular OWL/XML format. If ontology is stored in another format, external tools should be used first to convert it into OWL/XML.
- Check if the element is OWL RL. Only the OWL RL subset is converted, but it covers the big part of OWL Full. Elements not belonging to OWL RL will be skipped and should be converted into ES rules if needed manually or using other methods.
- Collect the non-OWL RL subset & Collect the OWL RL subset. After successful scanning the OWL/XML ontology, OWL RL elements are extracted for further processing. Non-OWL RL elements are separated for manual processing
- Transform to RDF triplets. The identified OWL RL elements are saved in the RDF triplet format (subject, predicate, object).
- Check if the element is a simple RDF triplet or a composite RDF triplet. The evaluation is performed if an RDF triplet can be unlinked from other RDF triplets.
- Save the element to a simple RDF triplet storage. The RDF storage stores RDS triplets that can be unlinked from other RDF triplets, e.g., the attributer type “symmetrical” is saved in the form of an RDF triplet (*?p rdf:type owl:FunctionalProperty*), where ?p is an attribute name. Such data is stored in the form of a text registry, having three fields.
- Save the element to a composite RDF triplet storage. Elements that cannot be unlinked are stored in storage for composite RDF triplets. for example, the ontology statement that all attributes ?p of object ?x are of ?y type: (*?x owl:allValuesFrom ?y*) (*?x owl:onProperty ?p*). Up to three linked RDF triplets can be stored.
- Analyze RDF triplets & Analyze composite RDF triplets. In this step, RDF triplets are transformed according to the conversion table, i.e., RDF triplet conversion templates. The main part of an RDF triplet is a predicate that can acquire values from a finite set of possible values because of the ontology formality feature. The remaining two triplet elements, subject and object, are specific concepts or attributes defined in the ontology. They are used to fill in the conversion template. In several specific cases (e.g., *owl:maxCardinality*), the template is modified.
- Make RIF rules. Templates are applied for RDF triplet conversion into RIF rules.

The proposed algorithm is provided in Fig. 2.4.

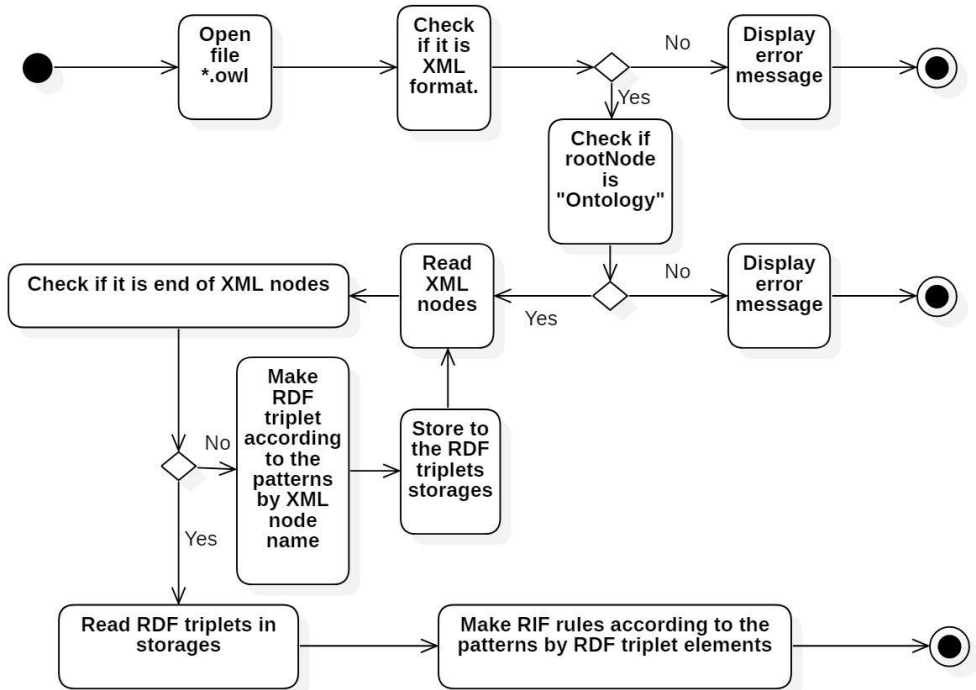


Fig. 2.4. Algorithm of OWL 2 RL transformation into RIF rules

Ontology scanning imports an ontology from an external .owl file in which the ontology is written in the OWL/XML format (the most common format). With an OWL ontology written in a different format, it must first be transformed into the OWL/XML format. There are many tools that can do this. During the ontology scan, all items found in the file are imported in sequence.

During the reading, the elements are immediately analyzed. This is because the elements of the OWL ontology would require an extremely complex structure. The program is designed to transform only OWL RL subset elements, but it covers a large part of OWL Full, so the ontology of any OWL subset can be specified for import, and unrecognized (non-OWL RL) elements will be skipped, thus transforming most of the ontology. For this reason, there is no clear structure that can accommodate the OWL Full ontology, and elements are analyzed individually or in groups if they are not separable.

When an OWL RL element is recognized, it is saved in the RDF triple format (subject, predicate, object), as it is easier to create rules from such a format. Inextricably linked elements may consist of more than one triple. Such elements are stored in the storage of composite RDF triples.

In the next step, the saved RDF triplets are transformed into RIF rules. This is done with pre-prepared templates that are populated with values obtained by analyzing the ontology. The resulting rules are fully compliant with the RIF Core format.

Scanning the ontology and transforming it into RIF rules is initiated from the user interface.

2.3.2. Software for Method Implementation

The formal algorithm in the form of pseudocode of OWL RL transformation into RIF rules is provided in Fig. 2.5.

```

READ selected_owl_file;
WHILE NOT End_of_xml_tokens DO
  GET xml_nodes
END DO
FOREACH xml_node in documentfigur_element DO //analyze each XML node
  IF node_name=="Declaration" THEN DO node_Declaration(node) //next step by the
node_name
  IF node_name=="DisjointClasses" THEN DO node_DisjointClasses(node)
//similar actions for other types of nodes. At all 18 types of nodes are used.
END DO

PROCEDURE node_Declaration (XmlNode node) //example of procedure by node_name;
  IF childNode_Name == "Class") THEN
    MAKE rdf_triple(attr.Value, "rdf:type", "owl:Class")
//making rdf triples by childNode names, at all 5 types of rdf triples for this class
  IF childNode_Name == "NamedIndividual") then
    MAKE rdf_triple(attr.Value, "rdf:type", "owl:NamedIndividual")
  END

DO rdf_to_rif //rdf triples are converting to RIF rules, using patterns

PROCEDURE rdf_to_rif(IN rdf_triples; OUT rule_1; OUT rule_2)
  READ rdf_triple
  IF rdf_triple_item2 == "rdf:type" THEN DO templateType(item, out rule1); //choose RIF
pattern by the RDF triple type
  IF rdf_triple_item2 == "owl:equivalentProperty": THEN DO
templateEquivalentProperty(item, out rule1, out rule2); //the similar for all rdf triple item types
END

PROCEDURE templateType(IN item, OUT string ans) //example of RIF rule pattern
  IF item == "owl:FunctionalProperty" THEN return_rule ("Forall ?y2 ?x ?y1 (?y1[owl:sameAs-
>?y2] :- And(?x[{o}->?y1] ?x[{o}->?y2]))", item.s1); //at all 20 similar RIF rules patterns

```

Fig. 2.5. Software pseudocode

While analyzing XML nodes, RDF triples are segregated. RDF triplets are created according to the predefined patterns given in Table 2.1:

Table 2.1. RDF triplets' patterns

IF	THEN
T (s ₁ , p ₁ , o ₁)	T (sr ₁ , pr ₁ , or ₁)
...	...
T (s _n , p _n , o _n)	T (sr _m , pr _m , or _m)

where each argument to the T predicate may be a variable or the literal value. RDF triplets created in such a way are later transformed into RIF rules.

```

Group (
  Forall ?v1 ... ?vo (
    sr1[p1->or1] :- And( s1[p1->o1] ... sn[pn->on] ))
    ...
  Forall ?vm ... ?vo (
    srm[pm->orm] :- And( s1[p1->o1] ... sn[pn->on] ))
  )

```

where ?v₁ ... ?v_o are the variables which occur in the rule.
Below, the sample transformation is provided.

Two following RDF triplets are extracted from the ontology:

```

attribute_1 rdf:type owl:SymmetricProperty
attribute_2 rdf:type owl:SymmetricProperty,

```

the template of symmetrical attributes is applied:

```

Forall ?x ?y (
  ?y[?p->?x] :- And(
    ?x[?p->?y] ))

```

the following RIF rules are obtained:

```

Forall ?x ?y (
  ?y[atributas_1->?x] :- And(
    ?x[atributas_1->?y] ))
Forall ?x ?y (
  ?y[atributas_2->?x] :- And(
    ?x[atributas_2->?y] ))

```

The transformation process is composed of two parts: the transformation of fixed rules (for the full list, see Annex A) and ontology rules obtained from its T-Box axioms. In the OWL 2 RL profile, both ontology axioms and additional logical rules can be stored, but in this research, only the subset of axioms is used as a suitable source for ES knowledge base rules.

Although an ontology typically has a very sophisticated structure, it can be easily represented in the form of RDF triples, as it was shown earlier.

Another issue of the ontology transformation is its correctness, which cannot be evaluated by the conversion program. The evaluation of ontology composing correctness is out of the scope of this research. It is assumed that ontologies used for experiments were correct and verified by other methods during their development process.

Some RIF rules (Fixed RIF rules) are obtained without analysis of ontology since their perception is a part of ES. Such rules are transformed into ES production rules only once and are loaded into the knowledge base. The sample of a fixed rule:

```
Forall ?c1 ?c2 (
  ?c1[rdfs:subClassOf->?c2] :- ?c1[owl:equivalentClass->?c2])
```

```
Forall ?x ?z ?y (
  ?x[owl:sameAs->?z] :- And(
    ?x[owl:sameAs->?y]
    ?y[owl:sameAs->?z] ))
```

```
Forall ?x ?y (
  rif:error() :- And(
    ?x[owl:sameAs->?y]
    ?x[owl:differentFrom->?y] ))
```

ES knowledge base rules are obtained by conversion of RIF rules that are automatically generated from the ontology as described earlier. As also stated earlier, some ontology elements, that were not converted automatically can be converted manually if they store some valuable information.

The final set of rules can be expressed by the equation:

$$R(RDF(O)) = \text{Fixed_rules} \cup \text{Ontology_rules}(RDF(O)), \quad (2.1)$$

where O is ontology, $RDF(O)$ RDF triplets of ontology O , $R(RDF(O))$ – RIF is a set of rules imported into the ES knowledge base.

The created program has five main parts: the user interface, the ontology scanning engine, the analysis engine, ontology alignment to the RDF triplet engine, and the transformation (into RIF rules) engine. The program class diagram is shown in Fig. 2.6.

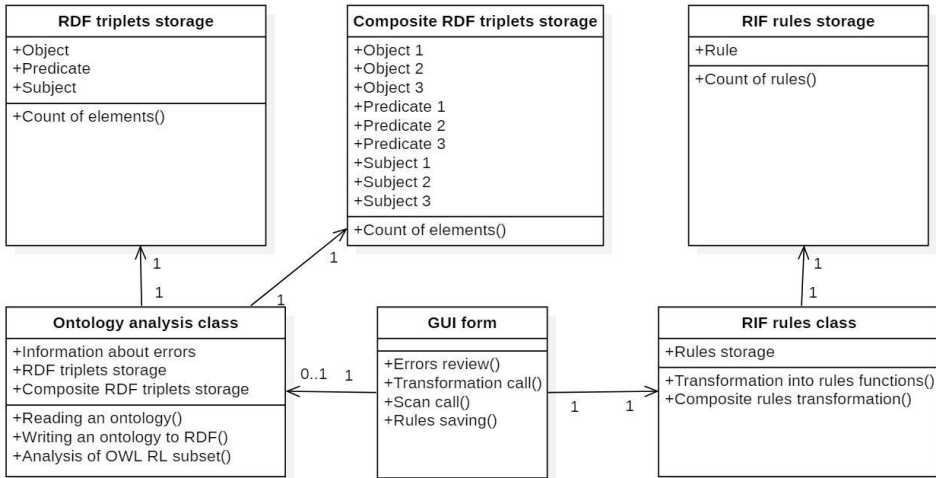


Fig. 2.6. Program class diagram

The program is composed of six main classes:

- GUI form. Program management. A user can call the ontology scanning, conversions into RIF rules, error review, and rules saving functions.
- Ontology analysis class. Responsible for the XML/OWL ontology scanning, identification of extractable elements, and their saving in the form of RDF triplets.
- RDF triplets' storage. Stores RDF triplets that can be unlinked from other RDF triplets.
- Composite RDF triplets' storage. Stores up to three linked together complex RDF triplets that cannot be unlinked.
- RIF rules class. Converts RDF triplets into RIF rules as defined.
- RIF rules storage. Stores RIF rules in the form of the presentational syntax that is later used for transformation into ES knowledge base rules.

The program execution sequence is shown in Fig. 2.7.

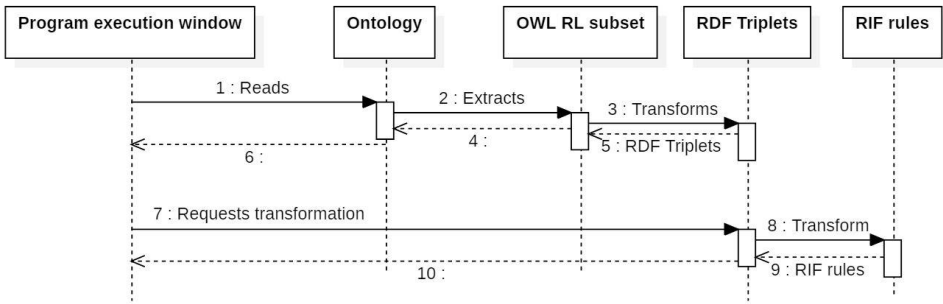


Fig. 2.7. Program execution sequence

All program control is performed from its main window, shown in Annex B Fig. B.1. All functionality is available from the main program window. The user must press the “Scan ontology” button and specify the path to the ontology file.

The ontology scanning engine imports the ontology from the external *.xml* file of the OWL/XML format. The program later performs the automatic scanning of elements, extraction of the OWL RL subset, and performs the conversion to RDF triplets. The results are shown in the main window. The user, after reviewing the intermedium data and the list of errors, can press the “Transform” button. After, RDF triplets are transformed to RIF rules and presented for user review. The user can copy or save them via the program interface into the specified file.

The received RDF triplets are displayed in the middle of the window. A part of the ontology which was not imported can be displayed by clicking “Options” - > “Show Errors”. An example of such errors is:

The ObjectPropertyDomain element or its internal elements do not conform to the OWL RL format.

The error list displays the names of items that do not match the OWL RL profile and other scan errors.

The user interaction is needed to initiate ontology imports and transformation processes. An example of a successfully transformed ontology into RIF rules is given below:

```

Forall ?x ?y (?x[:topObjectProperty->?y] :- And(?x[#negotiate_mean-
ing_of_ontology_definition_between->?y]))
Forall ?x ?y (?x[:topObjectProperty->?y] :- And(?x[#offer_service->?y]))
Forall ?x ?y (?x[:topObjectProperty->?y] :- And(?x[#reside_in->?y]))
Forall ?x ?y (?y[#access_resource->?x] :- And(?x[#accessed_by_service-
->?y]))
Forall ?x ?y (?y[#accessed_by_service->?x] :- And(?x[#access_resource-
->?y]))
Forall ?x ?y (?y[#controlled_by->?x] :- And(?x[#controls_entity->?y]))
  
```

```

Forall ?x ?y (?y[#controls_entity->?x] :- And(?x[#controlled_by->?y]))
Forall ?x ?y (?y[#used_by_application->?x] :- And(?x[#use_smart_entity-
>?y]))
Forall ?x ?y (?x[#hasAlignmentCellValue->?y] :- And(?x[#hasMeasureValue->?y]))

```

The comparative analysis of the risk program's window. The user can use the options menu to choose the results they want to view: the scanned ontology, created RIF rules or ontology elements that weren't transformed. All these results can be exported into a file.

The program was tested with several ontologies: W3C consortium test ontologies (Bock et al., 2012), cloud security ontology (Janulevičius et al., 2017), which is used in practice, and the information security standards ontology (Ramauskaitė et al., 2013). While W3C ontologies were developed by external parties, the information security standards and cloud security ontologies were developed at Vilnius Gediminas Technical University in accordance with formal OWL 2 RL rules defined by Cao et al. (2011) and OWL 2 Full defined by Bock et al. (2012). Their presentations and in-depth descriptions with verifications were done by their authors.

2.2.3. Example of Transformation

We have the following ontology elements (written in the OWL functional style syntax for simplicity):

```

Declaration(ObjectProperty(turiSūnų))
Declaration(ObjectProperty(turiTėvų))
InverseObjectProperties(turiSūnų, turiTėvų)

```

In the OWL/XML format, the same elements will look like this:

```

<Declaration>
  <ObjectProperty IRI="turiSūnų"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="turiTėvų"/>
</Declaration>
<InverseObjectProperties>
  <ObjectProperty IRI="turiSūnų">
  <ObjectProperty IRI="turiTėvų">
</InverseObjectProperties>

```

During the ontology scan, the program will store the following elements in the RDF triplet's structure:

```
("turiSünü", rdf:type, owl:ObjectProperty)
("turiTèvq", rdf:type, owl:ObjectProperty)
("turiSünü", owl:inverseOf, "turiTèvq")
```

The formed RDF triplets are transformed into such RIF rules:

```
Forall ?x ?y (?y["turiTèvq" -> ?x] :- And(?x["turiSünü" -> ?y]))
Forall ?x ?y (?y["turiSünü" -> ?x] :- And(?x["turiTèvq" -> ?y]))
```

They mean that if the element x has the attribute "turiSünü" and the element y belonging to it, then the element y has the attribute "turiTèvq" and the element belonging to it is equal to x. In other words, if Father John has a son Peter, then Son Peter has a Father John. The second rule is the opposite: if the son Peter has a father John, then the father John has a son Peter. RIF rules are read from the back: the part of the rule after the ":-" sign is a condition, and the conclusion is on the left.

A more complex example. Part of the ontology to be transformed into rules (declarations are skipped):

```
<SubClassOf>
  <Class IRI="#APISLayer"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="#isTheLayerAbove"/>
    <Class IRI="#AbstractionLayer"/>
  </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
  <Class IRI="#APISLayer"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="#isTheLayerAbove"/>
    <Class IRI="#Core_ConnectivityLayer"/>
  </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
  <Class IRI="#APISLayer"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="#isTheLayerBelow"/>
    <Class IRI="#IntegrationLayer"/>
  </ObjectSomeValuesFrom>
</SubClassOf>
<SubClassOf>
```

```

<Class IRI="#APISLayer"/>
<ObjectSomeValuesFrom>
  <ObjectProperty IRI="#isTheLayerBelow"/>
  <Class IRI="#MiddlewareLayer"/>
</ObjectSomeValuesFrom>
</SubClassOf>

```

Transformed into these RIF rules:

```

Forall ?v ?u (?u[rdf:type->#APISLayer] :- And(?u[#isTheLayerAbove->?v]
?v[rdf:type->#AbstractionLayer]))
Forall ?v ?u (?u[rdf:type->#APISLayer] :- And(?u[#isTheLayerAbove->?v]
?v[rdf:type->#Core_ConnectivityLayer]))
Forall ?v ?u (?u[rdf:type->#APISLayer] :- And(?u[#isTheLayerBelow->?v]
?v[rdf:type->#IntegrationLayer]))
Forall ?v ?u (?u[rdf:type->#APISLayer] :- And(?u[#isTheLayerBelow->?v]
?v[rdf:type->#MiddlewareLayer]))

```

W3c consortium test for “equivalentProperty” element (in original format):

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:first="http://www.w3.org/2002/03owlt/equivalentProperty/premises001#"
  xml:base="http://www.w3.org/2002/03owlt/equivalentProperty/premises001" >
  <owl:ObjectProperty rdf:ID="hasHead">
    <owl:equivalentProperty>
      <owl:ObjectProperty rdf:ID="hasLeader"/>
    </owl:equivalentProperty>
  </owl:ObjectProperty>
  <owl:Thing rdf:ID="X">
    <first:hasLeader>
      <owl:Thing rdf:ID="Y"/>
    </first:hasLeader>
  </owl:Thing>
</rdf:RDF>

```

Formed RIF rules:

```

Forall ?x ?y (?x[hasHead->?y] :- And(?x[hasLeader->?y]))
Forall ?x ?y (?x[hasLeader->?y] :- And(?x[hasHead->?y]))

```

2.4. Method Experiments and Verification

Verification and validation of the proposed method are important, and they are a part of this work. Verification and validation of this proposed method are divided into two parts: (1) verification that the generated rules meet specifications and fulfill their intended purpose, (2) validating the rules to determine whether they can be applied in real expert systems, dedicated to information security risk analysis, and satisfies the end-user requirements.

To verify that generated rules are correct and appropriate, several mechanisms were implemented in the software:

1. checks if the uploaded ontology format is supported,
2. RIF rules transformation is executed by prepared patterns, and if it doesn't match the pattern, such a rule is stored in the errors database. This ensures that generated RIF rules meet its specifications. Even more, successfully generated RIF rules and other results were reviewed by the group of experts.

The method validation part is given in Chapter 5, where the prototype of an expert system is demonstrated.

This part of the chapter presents the results of the ontologies transformation into RIF rules. All tested ontologies were relatively small (up to 1000 elements). The proportional size of an RL subset in ontologies used in tests was equal to 100 % in the case of W3C testing ontologies, and 65 % in the case of the information security standards ontology. The proportion of the OWL RL subset in the case of the cloud security ontology was not analyzed. The execution time has not exceeded 50 ms (test platform: Intel Core i7-4710HQ 2.50 GHz, 8 CPU, 16GB RAM, Windows 8.1 OS). Seventeen ontologies were tested in total, and the results are shown in Table 2.2.

Table 2.2. Transformation results

Ontology	The part of successfully transformed rules
W3C testing ontologies (15 in total), compatible OWL 2 RL	90 % (avg.)
Cloud computing security ontology	64 %
Information security standards ontology	60 %

As demonstrated, not all ontology elements were transformed. Some elements cannot be transformed into separate rules because they are just a part of another element or additional information. The best transformation result was achieved with test ontologies that are based on the OWL 2 RL profile. Specific security ontologies had shown worse results. In the case of Information security

standards ontology, data types *xsd.real* and *xsd.rational* have caused 15 % of unsuccessful conversions, 20 % were caused by *SubClassOf* elements, and 5 % by other *xsd* data types. The general method applicability was approved since the transformation of even 60 % can drastically decrease the knowledge base creation time.

Some successful transformations of the rules mostly depend on the ontology itself. When performing tests with W3C ontologies, ontologies corresponding to the OWL 2 RL profile were selected, resulting in most of the rules; but after taking the practical ontology, some elements did not correspond to a subset of OWL 2 RL, and some elements were not recognized.

The OWL 2 RL subset does not contain some popular OWL 2 Full items that have not been imported:

- *owl:minCardinality*
- *owl:minQualifiedCardinality*
- *owl:DisjointUnion*
- *owl:ReflexiveObjectProperty*
- *owl:topObjectProperty*
- *owl:bottomDatatypeProperty*
- *owl:topDataProperty*
- *owl:bottomDataProperty*

Also, OWL 2 RL is not allowed to use the *xsd: real* and *xsd: rational* data types. This comprised up to 15 percent of cases of untransformed rules. The problem could be solved by considering the amount of these elements in the imported ontology. If there are relatively few of them, such elements can be transformed manually, but some of their (e.g., *owl: topObjectProperty* *owl: bottomDatatypeProperty*) production rules have no equivalent and cannot be transformed. With large amounts of untransformed elements, new conversion table templates can be created while supplementing the import with these elements. It should be noted that the OWL 2 RL profile is of polynomial complexity (depending on the size of the ontology), and the knowledge base obtained from it is also of polynomial complexity. By supplementing the possible element types in any phase (before or after transformation), the estimate of the polynomial complexity may no longer be correct.

The OWL 2 RL profile restricts ontologies beyond the types of elements allowed. There are more constraints. For example, in the case of OWL 2 Full, the same resource (URI) can be both a concept and an individual, but OWL 2 RL does not allow it. Some elements can be used only partially, e.g., one of the popular elements *owl: maxCardinality* can only take values of 0 and 1, and data ranges

are limited to xsd data types (except xsd: real and xsd: rational). About 5 percent of elements not transformed for this reason were found during the experiments in practical ontologies. This problem can be solved in analogous ways: by transforming manually or by extending the transformation table.

Another important constraint is the prohibition of implicit elements. In the case of OWL 2 Full, inside elements (such as SubClassOf), it is possible to put internal non-express classes that are a union of other classes, an intersection, etc. OWL 2 RL does not allow this. During the experiment, this was one of the main problems (about 20 % of problem cases) that did not result in RIF rules because OWL ontologies that did not belong to the OWL 2 RL profile were imported (there are relatively few ontologies written in OWL 2 RL and efforts were made to test situations as close as possible to the actual use of the program). This problem can be solved by rearranging the ontology by giving the implicit classes names and declarations.

During the experiments, W3C ontologies were transformed just for method testing purposes. Other transformed ontologies are applied in practice in the field of information security. After such ontologies were transformed, 601 RIF rules (out of 1002 elements) were obtained from the information security standards ontology in an automatic way, and 61 RIF rules (out of 95 elements) were obtained from the cloud computing security ontology. Thus, a total of 662 rules were generated automatically, 435 rules had to be manually transformed.

For further validation, the generated rules were integrated into the JESS-based prototype risk analysis ES, adapted for SMEs, which is presented in Chapter 5.

2.5. Conclusions of Chapter 2

1. This chapter proposed the method that allows automatic transformation of existing information security ontologies into a ruleset of the expert system's knowledge base. The transformation of the information security standards ontology into the JESS ES was implemented in several steps with the help of the developed tool. The OWL 2 RL subset of ontology elements was segregated into RDF triplets, which were later transformed into the RIF format that can be easily converted into JESS knowledge bases rules with the help of an external tool. The method supports the most popular OWL/XML format. Both simple (unlinked) and complex (linked) OWL RL elements can be transformed. The greatest advantage of the method is that although an ontology typically has a very sophisticated structure, it can be easily represented in the form of RDF triplets.

2. The conversion test with different ontologies has shown a 60–90 % conversion success rate, which proves general method applicability since the automatic generation of even a part of the expert system's knowledge base can decrease the general expert system development price. The obtained results have shown a high correlation rate because it allows transforming a greater part of the ontology into rules of an expert system's knowledge base automatically, while another smaller ontology part can be transformed manually. This makes the process of the ontology transformation into expert system rules more efficient compared to the manual transformation.
3. During the experiments, 15 W3C ontologies were transformed just for method testing purposes. The other two transformed ontologies are applied in practice in the field of information security. After such ontologies were transformed, 601 RIF rules (out of 1002 elements) were automatically obtained from the information security standards ontology, and 61 RIF rules (out of 95 elements) were obtained from the cloud computing security ontology. Thus, a total of 662 rules were generated automatically, and 435 rules needed to be manually transformed. Some elements cannot be transformed into separate rules because they are just a part of another element or additional information.

Integration of Data on Regional Malware Distribution into the Expert System's Knowledge Base for Geographically Adapted Risk Evaluation

This chapter presents a novel method for an automated generation of an expert systems' knowledge base established on the integration of data on regional malware distribution from the cyberthreat real-time map providing current information on newly discovered threats. It automatically collects the necessary information from websites to be used later in a relevant risk calculation. This chapter presents the method implementation, which includes not only the knowledge base formation but also an experiment with an expert system. It was created using the JESS expert system shell. The information security risk assessment was performed according to the OWASP risk assessment methodology, considering the location of the organization and the prevalent malware in that area.

The proposed method and results were published in an international journal (Vitkus, Jezukeviciute, et al., 2020).

3.1. Integration of Data on Regional Malware Distribution from the Cyberthreat Real-Time Map into the Knowledge Base

This chapter proposes a novel approach to the knowledge base formation for expert systems compared to the traditional one that requires real experts and has a long process of the knowledge base formation, which is sometimes expensive (Kireeva et al., 2019b). It is suitable in cases where long-lasting data can be used but not in information security risk assessment. A knowledge base used in information security risk assessment should reflect a constantly changing threat landscape and evaluate the risk likelihood depending on the region and other factors.

The proposed method is based on the acquisition of data from trusted and open websites providing current information on newly discovered threats. The proposed method was implemented by data acquisition from the cyberthreat real-time map website by Kaspersky Lab (*Cyber-Threats Real-Time Map*, n.d.).

Automatically formed facts were imported into an expert system. The expert system was designed to perform the information security risk assessment according to the OWASP methodology (Owasp, 2013). The formed knowledge base was implemented on the Java Expert System Shell (JESS) (E. Friedman-Hill, 2007). The acquired data was automatically imported into JESS using the XML format. It must be stated that automatically acquired data has formed only a part of the knowledge base, i.e., it was integrated into the existing knowledge base and was used to estimate the malware likelihood used in risk calculation in the specific region.

This work presents an automated expert systems knowledge base formation method based on the acquisition of data from trusted and open websites providing current information on newly discovered threats by reading their HTML code and extracting information between tags. HTML was chosen because it is widely used (Saini et al., 2008) and structured, so it can be easily transformed into another format.

The proposed method is shown in Fig. 3.1. It has several steps. The first step is data extraction, which includes:

- sending HTTPS requests to the specified WEB page,
- obtaining the HTTPS response from the WEB server,
- extracting useful data about malware from HTML tags and stored to SQL DB.

The second step after data extraction is data conversion: all data in the SQL DB are converted to the XML format.

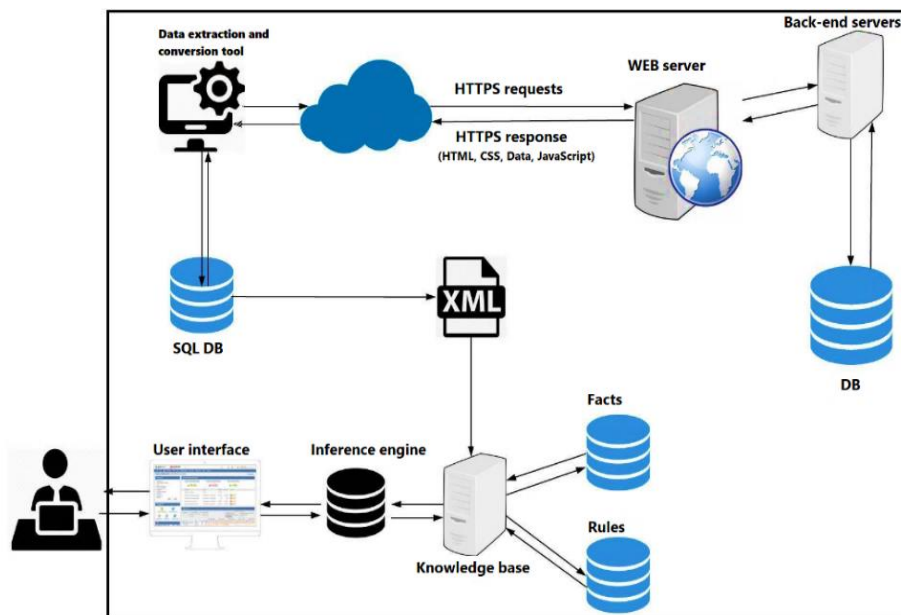


Fig. 3.1. Architectural view of the method implementation

And the last step is the knowledge base formation: the created XML file is imported into the existing expert system.

3.1.1. Acquisition of Data from Websites

The method is implemented with a tool created in PHP language, which addresses the specified website by sending the HTTPS request. The algorithm for data acquisition from websites is given in Fig. 3.2.

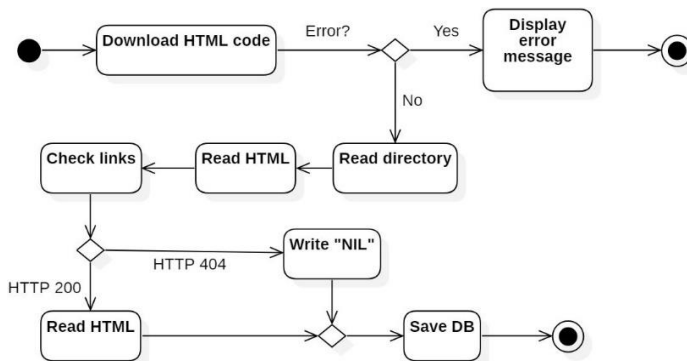
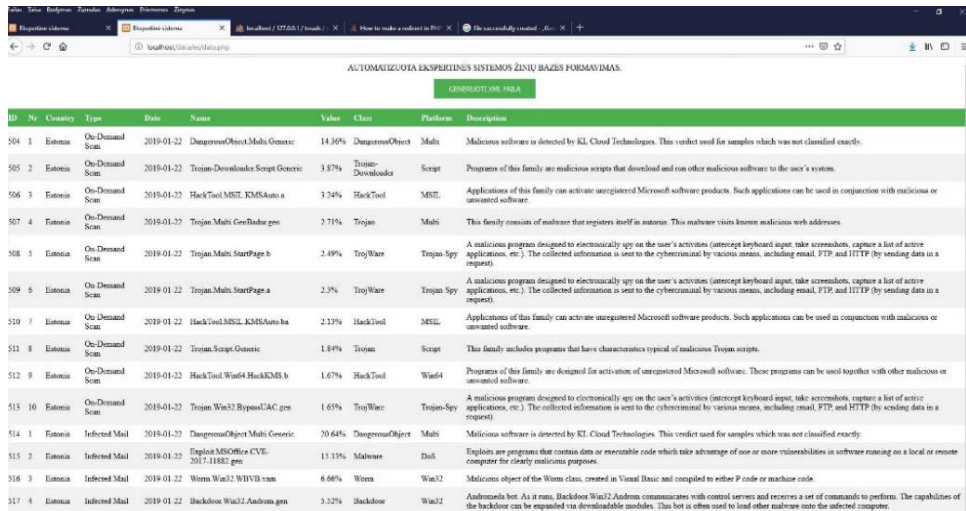


Fig. 3.2. Algorithm for data acquisition from websites

When a web server responds to the query, the tool reads the HTML code and processes the information between the specified tags. The collected data is stored in the SQL database. This way, the needed knowledge is automatically extracted from websites and can be later used in relevant risk calculations. An example of the automatically extracted data is shown in Fig. 3.3.



ID	No.	Country	Type	Date	Name	Value	Class	Platform	Description
504	1	Estonia	On-Demand Scan	2019-01-22	DangerousObject.Multi.Generic	14.36%	DangerousObject	Multi	Malicious software is detected by K1 Cloud Technologies. This verdict used for samples which was not classified exactly.
505	2	Estonia	On-Demand Scan	2019-01-22	Trojan.Downloader.Script.Generic	3.87%	Trojan-Downloader	Script	Programs of this family are malicious scripts that download and run other malicious software to the user's system.
506	3	Estonia	On-Demand Scan	2019-01-22	HackTool.MSI2_KMSAuto.a	3.24%	HackTool	MSI2	Applications of this family can activate unregistered Microsoft software products. Such applications can be used in conjunction with malicious or unwanted software.
507	4	Estonia	On-Demand Scan	2019-01-22	Trojan.Multi.Gen.Bodnar.gen	2.71%	Trojan	Multi	This family consists of malware that registers itself as malware. This malware visits known malicious web addresses.
508	5	Estonia	On-Demand Scan	2019-01-22	Trojan.Multi.StartPage.b	2.49%	Trojan-Ware	Trojan-Spy	A malicious program designed to electronically spy on the user's activities (intercept keyboard input, take screenshots, capture a list of active applications, etc.). The collected information is sent to the cybercriminal by various means, including email, FTP, and HTTP (by sending data in a request).
509	6	Estonia	On-Demand Scan	2019-01-22	Trojan.Multi.StartPage.a	2.3%	Trojan-Ware	Trojan-Spy	A malicious program designed to electronically spy on the user's activities (intercept keyboard input, take screenshots, capture a list of active applications, etc.). The collected information is sent to the cybercriminal by various means, including email, FTP, and HTTP (by sending data in a request).
510	7	Estonia	On-Demand Scan	2019-01-22	HackTool.MSI2_KMSAuto.ba	2.13%	HackTool	MSI2	Applications of this family can activate unregistered Microsoft software products. Such applications can be used in conjunction with malicious or unwanted software.
511	8	Estonia	On-Demand Scan	2019-01-22	Trojan.Script.Generic	1.84%	Trojan	Script	This family includes programs that have characteristics typical of malicious Trojan scripts.
512	9	Estonia	On-Demand Scan	2019-01-22	HackTool.Win64.HackKMS.b	1.67%	HackTool	Win64	Programs of this family are designed for activation of unregistered Microsoft software. These programs can be used together with other malicious or unwanted software.
513	10	Estonia	On-Demand Scan	2019-01-22	Trojan.Win32.BypassUAC.gen	1.65%	Trojan-Ware	Trojan-Spy	A malicious program designed to electronically spy on the user's activities (intercept keyboard input, take screenshots, capture a list of active applications, etc.). The collected information is sent to the cybercriminal by various means, including email, FTP, and HTTP (by sending data in a request).
514	1	Estonia	Inferred Mail	2019-01-22	DangerousObject.Multi.Generic	20.64%	DangerousObject	Multi	Malicious software is detected by K1 Cloud Technologies. This verdict used for samples which was not classified exactly.
515	2	Estonia	Inferred Mail	2019-01-22	Exploit.MSOffice.CVE-2017-11882.gen	15.13%	Malware	Exploit	Exploits are programs that contain data or executable code which take advantage of one or more vulnerabilities in software running on a local or remote computer for clearly malicious purposes.
516	3	Estonia	Inferred Mail	2019-01-22	Worm.Win32.WIVV.vnm	6.66%	Worm	Win32	Malicious object of the Worm class, created in Visual Basic and compiled to either P code or machine code.
517	4	Estonia	Inferred Mail	2019-01-22	Backdoor.Win32.Android.gen	3.32%	Backdoor	Win32	Android.mal bot. As it runs, Backdoor.Win32.Android communicates with control servers and receives a set of commands to perform. The capabilities of the backdoor can be expanded via downloadable modules. This bot is often used to load other malware onto the infected computer.

Fig. 3.3. Example of automatically extracted data

Automatically collected information about malware is stored in the SQL database. It must be transformed into a format suitable for expert systems to be used by them. An example of the database shown in the figure is a source for the formation of knowledge base rules.

3.1.2. Knowledge Base Formation and Import into an Expert System

The knowledge base formation is done by converting extracted data from SQL DB to XML format and importing it to JESS. An example of data conversion is shown in Fig. 3.4.

This method utilizes the prepared knowledge base structure, which accepts automatically generated facts:

(deftemplate question
(slot id)
(slot nr)
(slot country)

```

(slot type)
(slot date)
(slot name)
(slot value)
(slot class)
(slot platform)
(multislot description)
)

```

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<fact-list>
  <fact>
    <name>MAIN::question</name>
    <slot>
      <name>id</name>
      <value type="INTEGER">503</value>
    </slot>
    <slot>
      <name>nr</name>
      <value type="INTEGER">1</value>
    </slot>
    <slot>|
      <name>country</name>
      <value type="STRING">Estonia</value>
    </slot>
    <slot>
      <name>type</name>
      <value type="STRING">Botnet activity</value>
    </slot>
    <slot>
      <name>description</name>
      <value type="INTEGER"> nil</value>
    </slot>
  </fact>

```

Fig. 3.4. Example of data conversion to XML format

The formed knowledge base was implemented on the Java expert system shell (JESS). All collected data was automatically imported into JESS using the

XML format. Fig. 3.5 demonstrates an example of the generated knowledge base imported into the JESS expert system prototype.

```
(deftemplate question
  (slot id)
  (slot nr)
  (slot country)
  (slot type)
  (slot date)
  (slot name)
  (slot value)
  (slot class)
  (slot platform)
  (multislot description))

(load-facts book.xml)
(facts)
```

f-134 (MAIN::question (id 637) (nr 10) (country "Lithuania") (type "Local infections") (date "2019-01-22") (name "Trojan.Win32.Hosts2.gen") (value "2%") (class "TrojWare") (platform "Trojan-Spy" (description "Known as a hazard to the security of computers to largely slow down computer performance and drive computer abnormal in work when it is downloaded"))

f-135 (MAIN::question (id 638) (nr 1) (country "Lithuania") (type "On-Demand Scan") (date "2019-01-22") (name "DangerousObject.Multi.Generic") (value "7.95%") (class "DangerousObject") (platform "Multi" (description "Malicious software is detected by KL Cloud Technologies. This verdict used for samples which was not classified exactly."))

Fig. 3.5. Example of the generated knowledge base

This figure shows an example of formed JESS knowledge base facts. An example of a successful knowledge base import into an expert system is provided in Annex B Fig. B.2.

It must be stated that automatically acquired data has formed only a part of the knowledge base, i.e., it was integrated into the existing knowledge base and was used only for estimation of malware likelihood used in risk calculation in the specific region. The prototype interacts with the user via a dialogue session. Risk identification is made by matching virus distribution platforms and platforms used in the organization. Also, the location of the organization and the prevalence of

viruses in that area are evaluated. If an organization is operating in a country where the prevalence of the virus is recorded, the potential risk is calculated.

An example of the HTML transformation to JESS facts is given in Fig. 3.6. This example shows HTML tags, which were transformed, and the brown box gives an example of a JESS fact as a transformation result.

```
<td>Country</td>
<td>% Percent of users*</td>
<td>1</td>
<td>Lithuania</td>
<td>22.51</td>

<div class="stats_panel" id="country_stats_list">
  <h2>Top - On-Demand Scan IN THE LAST WEEK</h2>
  <table id="country_stats_threats_list" cellpadding="0" cellspacing="0"><tr class
    ="threats_link" data-link=
    ="https://threats.kaspersky.com/en/threat/Trojan.Multi.Hosts2.gen"><td class=
    ="element_index">1</td><td class="element_name">Trojan.Win32.Hosts2.gen</td><td
    class="element_value">22.51% </td>

    <td class="cell_one cell_one_icone">Detect Date</td>
    <strong>02/02/2019</strong>
    <td class="cell_one cell_one_icone">Class</td>
    <a href="https://threats.kaspersky.com/en/class/Trojan/" class="btn-link">
      Trojan
    <td class="cell_one cell_one_icone">Platform</td>
    <a href="https://threats.kaspersky.com/en/platform/Multi/" class="btn-link">
      Multi
    </td>

    <td class="cell_one cell_one_icone">Description</td>
    <td class="cell_two ">
    <p>Malware of this family is installed on browsers deceptively after the user
    visits fraudulent or advertising resources. This malware displays advertising
    messages even if a browser is inactive.</p>
  </td>
</div>
```



JESS fact:

f-1 (MAIN::question (id 1) (nr 1) (country "Lithuania") (type "On-Demand Scan") (date "2019-02-02") (name "Trojan.Win32.Hosts2.gen") (value "22.51%") (class "Trojan") (platform "Multi") (description "Malware of this family is installed on browsers deceptively after the user visits fraudulent or advertising resources. This malware displays advertising messages even if a browser is inactive."))

Fig. 3.6. Example of the HTML transformation to JESS fact

Fig. 3.7 gives an example of one rule transformation from the website view into a JESS knowledge base fact.

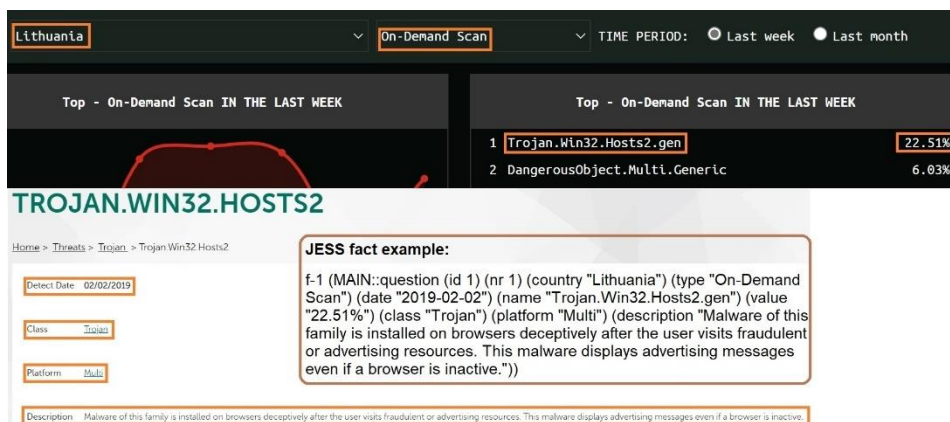


Fig. 3.7. Example of the transformation of the JESS knowledge base fact from the website view

Orange boxes in the figure show elements transformed into the JESS fact. The result of the formed JESS fact is given in a brown box.

3.2. Method Experiments and Risk Analysis Based on Regional Malware Distribution

The proposed method was tested on data acquisition from the Cyberthreat real-time map website by Kaspersky Lab (Cyber-Threats Real-Time Map, n.d.). This site was chosen because it contains up-to-date information about known malicious code and its distribution location in approximately 200 locations in total. In this work, information about 3 Baltic region countries was used: Lithuania, Estonia, and Latvia. The extracted information contained the following information about malware: name, type, the prevalence in the selected country, spreading platform, class, description, system, how it works, etc.

While implementing the proposed method for the experiments, a data extraction and conversion tool was created to get data from HTML, format the knowledge base, and convert it into the XML format. Also, a prototype of an expert system using a knowledge base with automatically acquired data was developed. The prototype was designed to perform the SME's information security risk assessment.

During the experiment, 225 web tags were scanned in total, and 201 entries of the knowledge base were formed automatically. Use case diagram of the method is shown in Fig. 3.8.

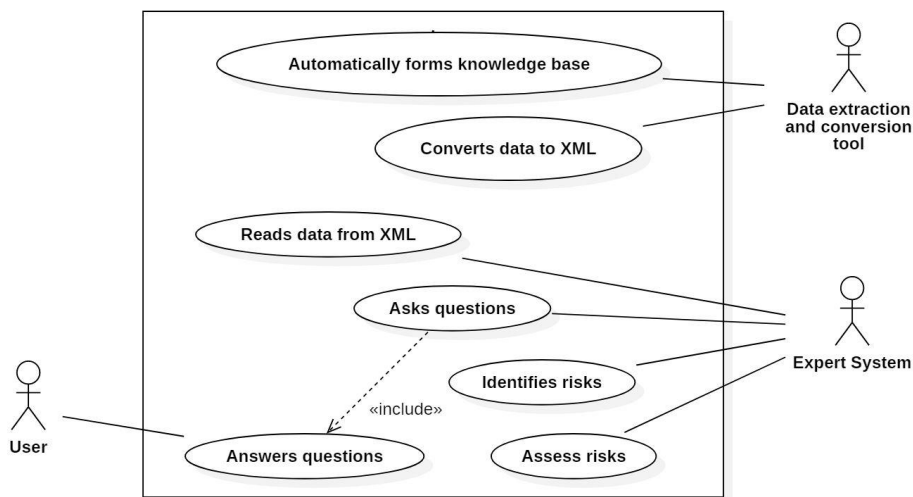


Fig. 3.8. Method use case diagram

The proposed method has demonstrated a principal possibility of automatic data extraction and import to the expert system knowledge base. Risk assessment, using the automatically formed knowledge base, was conducted according to the OWASP methodology. According to user answers, technical and business impacts were calculated in values from 0–9. The likelihood was evaluated according to the comparison of enterprise information system platforms and the malware platform.

Finally, the risk is calculated by the formula: $RISK = LIKELIHOOD * IMPACT$. An example of the risk calculation is given in Fig. 3.9.

Technical Impact MEDIUM: 5.0	

Business Impact MEDIUM: 8.0	

Technical risk:	20
Business risk:	32
Platform:	HTML
Percentage of spread:	4.02
Country:	Lithuania
Virus Name:	Hoax.HTML.FraudLoad.m
Type:	Web threats
Class:	Hoax
Description:	This family includes HTML pages that require the user to download

Fig. 3.9. Example of the risk calculation

The impact is evaluated by the business criteria; likelihood is a percentage of the malware spread in a specific region. Fig. 3.10 gives an example of one different malware risk value spread by regions, which is different in a specific region.

Impact level: 8.0

Risk level (LT):	160
Risk level (LV):	97
Risk level (EST):	104
Platform:	Multi
Percentage of spread (LT):	20.1
Percentage of spread (LV):	12.1
Percentage of spread (EST):	13.04
Virus Name:	DangerousObject.Multi.Generic
Type:	On-Access Scan
Class:	DangerousObject
Description:	Malicious software is detected by KL Cloud

Fig. 3.10. Example of malware risk value in Lithuania, Latvia, and Estonia

The given example shows that malware “DangerousObject.Multi.Generic” is spread unevenly in different regions. So, the risk value related to this malware is different even if the impact value is the same.

The main advantages of the proposed method are the possibility to keep the knowledge base up to date with relevant information on current threats, integrate information from multiple sources of information, and to minimize the workload while creating the knowledge base of an expert system. Limitations of the method are related to the changing structure of websites providing data and the need to update the tool code manually. Consequently, further research is to be concentrated on tool adaptability for website structure.

3.3. Conclusions of Chapter 3

- 1. A novel automated expert systems’ knowledge base formation method based on the data acquisition from trusted and open websites providing current information on newly discovered malware and its geographic distribution was proposed. The method is based on scanning the HTML code, importing processed tag information into the database with further conversion into XML code, and integration into the knowledge base of a developed JESS-based expert system for geographically adapted malware risk assessment.

2. The Kaspersky Lab's cyberthreat real-time map website was chosen as a data source for the method implementation tests. Data on geographical malware distribution were collected for three countries: Lithuania, Latvia, and Estonia. The following data on cyber threats were automatically collected: malware description, related vulnerabilities, the prevalence of malware in the selected region, the malware spreading platform, and the typical impact on the system.
3. The method tests on the developed JESS-based expert system have shown that automatically extracted data from the websites can improve the quality and expediency of risk assessment and provide geographically adapted results. The proposed method differs from earlier methods by the ability to dynamically update the knowledge base rules and facts, thus reacting to the changing cyberthreat landscape, automatically presenting the emerging threats in a risk assessment by linking them to a geographic company location. During the experiment, 225 web tags were scanned in total, and 201 entries of the knowledge base were formed automatically.

Method for Attack Tree Data Transformation and Import into IT Risk Analysis Expert Systems

This chapter proposes a novel method of converting attack trees into a format usable by expert systems to utilize existing attack tree repositories in facilitating information and IT security risk analysis. The method performs the attack tree translation into the JESS format by consistent application of the ATTop, the software bridging tool that enables automated analysis of attack trees using a model-driven engineering approach, translating attack trees into the eXtensible Markup Language (XML) format, and the newly developed attack trees into the Expert System (ATES) program, performing further XML conversion into the JESS-compatible format.

The proposed method and results were published in an international journal (Vitkus, Salter, et al., 2020).

4.1. Automation of Converting Information from Attack Trees into Expert System's Knowledge Base Rules

This chapter proposes a novel approach to knowledge base formation for expert systems dedicated to IT security risk analysis and using attack trees as a source of information. The method proposed in this work is novel and exclusive because of the approach based on which attack trees are transformed into rules of an expert system's knowledge base. Earlier approaches attempted to use attack tree data directly without the transformation. The proposed approach provides a unique possibility to generate an integral knowledge base of an expert system by integrating different attack trees taken from various sources. Furthermore, the generated rules can be imported into the existing knowledge base, where knowledge of other areas of information security risks is already systemized. While performing the transformation, information on attack directions, probabilities, impact, and possible countermeasures is imported, thus providing the possibility to use an expert system for decision making and making recommendations on risk management. The research aims to improve the methods for the creation of a knowledge base for the eventual purpose of assisting with risk analyses on an information system or set of systems.

Attack trees can assist in the IT risk analysis process by providing a structure to contemplate an attack against a system, i.e., attack trees provide a formal, methodical way of describing the security of systems, based on varying attacks, that can be very useful while evaluating possible threats and their probabilities during the risk assessment. The primary purpose of attack trees is to model security threats, represent attacks against a system, and analyze attack vectors. Some researchers have demonstrated (D. Kim et al., 2019; Kornecki & Liu, 2013; Qiu et al., 2019; Rios et al., 2020) that attack trees can be used as a supportive tool in the field of defense, vulnerabilities detection. Attack trees are also successfully used in information security risk analysis (Ingoldsby, 2013), the design process of security and defense systems (Löhner, 2018), and their analysis (Kordy et al., 2013). The versatility of attack trees and wide areas of their application has allowed assuming that attack trees can be considered a reliable source of information that can be applied to increasing the efficiency of expert systems. This chapter proposes the idea of using attack trees for IT security risk analysis by converting them into ES knowledge base rules.

By developing a process that streamlines the risk assessment via expert systems, risk analysis can be made more affordable for small and medium enterprises and individuals while simultaneously increasing the accuracy of the risk assess-

ment (Gadyatskaya, Harpes, et al., 2016; Janulevičius & Čenys, 2014) for everyone, especially SMEs that cannot afford to hire high-quality security experts. Under current methods, a cybersecurity professional, when using an attack tree, will be required to painstakingly proceed through several attack trees to perform a risk analysis for a system or to base their decision on a subjective valuation. This is a task that can take a considerable amount of time and effort as an expert will have to spend their time going through every attack route possible within the attack trees, some of which can have thousands of nodes (Shostack, 2014). Solving the problem of converting attack trees to formats that an expert system can utilize will help reduce the time costs associated with performing risk assessments. The creation of a knowledge base provides a higher level of information security since countless examples of attack trees already exist with varying levels of detail and complexity, targeted against different information systems and platforms. Utilizing an expert system for this process would also ensure accuracy, reliability, and repeatability for conducted risk analyses. By converting attack trees to formats usable by expert systems, the benefits of attack trees and expert systems can be combined and multiplied. Moreover, this method allows the use of existing resources and repositories of attack trees to assist in building a wide knowledge base.

Automating the conversion of attack trees to a format that can be used by expert systems is relevant and essential because it will allow small companies and individuals to perform risk assessments at a level comparable to an assessment conducted by someone with the resources of large companies (Gadyatskaya, Harpes, et al., 2016). Findings on attack tree data transformation and the import of risk analysis expert systems into it can be applied to significantly reduce the amount of time, effort, and monetary investment required for the risk analysis of a given company or agency. Additionally, this will allow organizations with limited security financing to prioritize the defense of the most likely places where an attack could occur.

4.1.1. Attack Tree Preparation for Transformation

The proposed method is a process that utilizes two software bridging programs together to implement the automation of converting information from attack trees into rules of an expert system's knowledge base. This method is implemented via a python-based program named ATES, which will take output in the Universal Metamodel for Attack Trees (UATS) format from the ATTop program mentioned above and will create ES knowledge base rules for the output files. The output of ATES is either one or two files depending on the user selection, which is readable by the JESS language. Version one outputs a singular file containing the structure for the JESS expert system and the attack tree data, while version two outputs two

files, one containing the structure for the JESS expert system and the other containing the attack tree data.

4.1.2. Converting UATS to Expert System Rules

The general view of the proposed method is presented in Fig. 4.1.

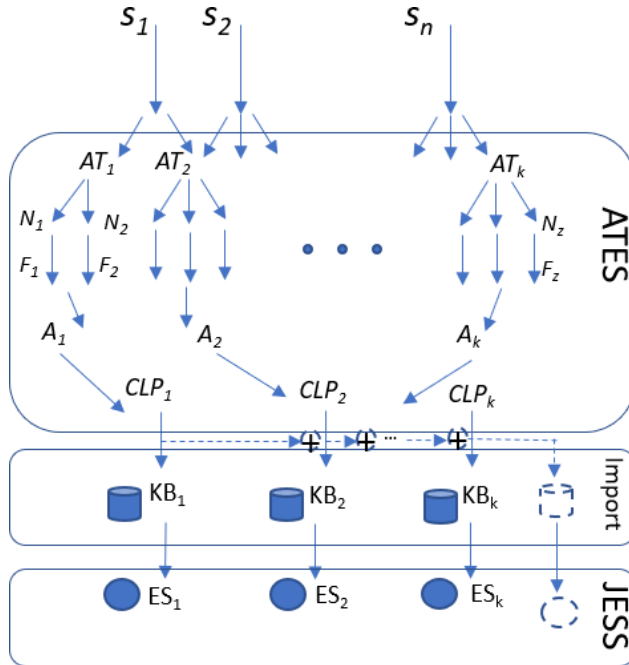


Fig. 4.1. General view of the proposed method

The proposed method is based on data transformation from attack trees ($AT_1 \dots AT_k$) collected from different sources ($S_1 \dots S_n$) into ES knowledge base facts. Nodes and leaves ($N_1 \dots N_z$) of the attack trees are transformed into facts ($F_1 \dots F_z$), while the root of the attack tree is used to generate the attack description ($A_1 \dots A_k$). Every transformed attack tree is saved in the form of CLIPS rules in a newly generated file ($CLP_1 \dots CLP_k$). Every generated CLP file is formed as a knowledge base of an expert system ($KB_1 \dots KB_k$) that can be later imported into the unique expert system ($ES_1 \dots ES_k$) or can be merged with an existing knowledge base (dotted notation in Fig. 4.1).

The proposed method does not depend on the source of attack trees, and it does not limit the amount of attack trees, i.e., S as the source of all attack trees, can be formed of several different sources.

$$S = \{S_1, S_2, \dots, S_n\}. \quad (4.1)$$

Each source can have several attack trees (AT).

$$S_1 = \{AT_1, AT_2, \dots, AT_l\}; \quad (4.2)$$

$$S_2 = \{AT_{l+1}, AT_{l+2}, \dots, AT_j\}; \quad (4.3)$$

...

$$S_n = \{AT_{y+1}, AT_{y+2}, \dots, AT_k\}. \quad (4.4)$$

Consequently, attack trees from different sources can be joined, thus forming the set of attack trees AT.

$$AT = \sum_{i=0}^n (S_i); \quad (4.5)$$

$$AT = \{AT_1, AT_2, \dots, AT_k\}. \quad (4.6)$$

Each attack tree can have a different depth and structure with a different number of leaves (marked as N). Still, every attack tree has a root leaf, called a Root node (marked as R).

$$AT_1 = \{R_1, N_1, N_2, \dots, N_o\}; \quad (4.7)$$

$$AT_2 = \{R_2, N_{o+1}, N_{o+2}, \dots, N_p\}; \quad (4.8)$$

...

$$AT_k = \{R_k, N_{x+1}, N_{p+2}, \dots, N_z\}. \quad (4.9)$$

Joint leaves and root nodes of all attack trees result in the set of attack tree elements NR .

$$NR = \sum_{i=0}^k (AT_i); \quad (4.10)$$

$$NR = \{R_1, R_2, \dots, R_k, N_1, N_2, \dots, N_z\}. \quad (4.11)$$

Transformation: Every element in set AT is transformed into elements in the set A :

$$AT\{AT_1, AT_2, \dots, AT_k\} \Rightarrow A\{A_1, A_2, \dots, A_k\}, \quad (4.12)$$

i.e., the attack tree is transformed into separate attacks in the knowledge base. Since the AT set is formed of several elements, every element should be transformed separately. The set of attack tree elements NR is transformed into the set of knowledge base facts and aims FG :

$$NR\{R_1, R_2, \dots, R_k, N_1, N_2, \dots, N_z\} \Rightarrow FG\{G_1, G_2, \dots, G_k, F_1, F_2, \dots, F_z\}. \quad (4.13)$$

In other words, every attack set A is formed of the attack aim G and facts F :

$$A_1 = \{G_1, F_1, F_2, \dots, F_o\}; \quad (4.14)$$

$$A_2 = \{G_2, F_{o+1}, F_{o+2}, \dots, F_p\}; \quad (4.15)$$

...

$$A_k = \{G_k, F_{x+1}, F_{x+2}, \dots, F_z\}. \quad (4.16)$$

After the transformation is performed, data must be prepared in a format suitable for the expert system. The author used the rules of the C Language Integrated Production System (CLIPS). Therefore, the CLP file was formed composed of three main elements:

- tree template (TT) – information about the converted Attack tree (AT name and root node (attack goal) name);
- node template (NN) – predefined structure to work with generated facts from attack trees;
- facts about the attack, generated in the transformation process, i.e., CLP if composed of the following elements:

$$CLP_1 = \{TT, NT, A_1\} = \{TT, NT, G_1, F_1, F_2, \dots, F_o\}; \quad (4.17)$$

$$CLP_2 = \{TT, NT, A_2\} = \{TT, NT, G_2, F_{o+1}, F_{o+2}, \dots, F_p\}; \quad (4.18)$$

...

$$CLP_k = \{TT, NT, A_k\} = \{TT, NT, G_k, F_{x+1}, F_{x+2}, \dots, F_z\}. \quad (4.19)$$

Rules generated using this automatic method can be used separately or merged into a single rule:

$$CLP = \sum_{i=0}^k (CLP_i); \quad (4.20)$$

$$CLP = \{TT, NT, G_1, G_2, \dots, G_k, F_1, F_2, \dots, F_z\}. \quad (4.21)$$

Fig. 4.2 depicts an activity diagram of the attack tree data proceeding by the methodology from creation as an attack tree to the final transformation to the rules of the JESS expert system's knowledge base.

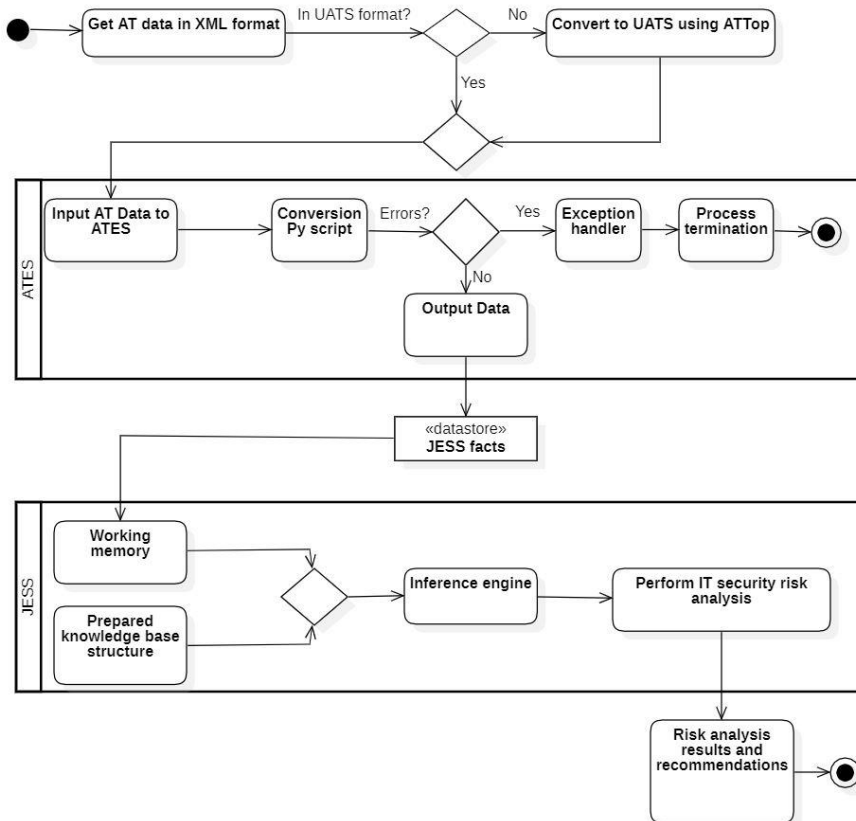


Fig. 4.2. Activity diagram of the proposed method

The main steps presented in Fig. 4.2 are described below.

- Get AT data in the XML format. The initial transformation of AT into the XML format is performed due to its versatility. The XML UATS format enables the use of a larger variety of AT sources, so the tool is made to work with XML UATS from the beginning. Examples of AT in XML UATS and graphical formats are given in Fig. 4.3.

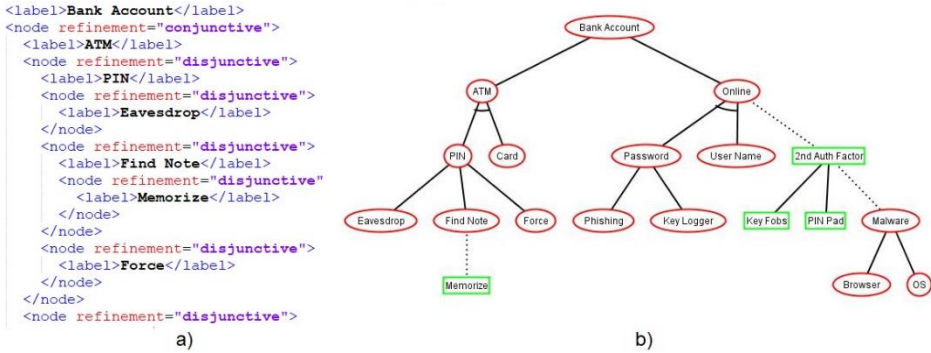


Fig. 4.3. Examples of attack trees: a) AT in the XML UATS format; b) AT in the graphical format

- Convert to UATS using ATTop. The attack tree can be directly exported to XML UATS or converted from various formats. The tool proposed to be used in this step is ATTop (ATTop, n.d.).
- Input AT Data to ATES. ATES is the created python-based software for the practical implementation of the method that performs the conversion of UATS into the JESS format.
- Conversion Python script is the conversion engine that converts XML UATS to CLP rules and DAT facts by applying defined patterns. Examples of resulting ES knowledge base facts are presented in Fig. 4.4.

```

(MAIN::tree (tname "Accessing a bank's gold") (rootnode "Gain access to vault gold"))
(MAIN::node (id 0) (name "Gain access to vault gold") (parent nil) (connector AND) (cost 0) (time 0) (exploited FALSE) (probability 0.0) (difficulty 0) (countermeasure FALSE) (children 1 8))
(MAIN::node (id 1) (name "Get inside bank") (parent 0) (connector OR) (cost 0) (time 0) (exploited FALSE) (probability 0.0) (difficulty 0) (countermeasure FALSE) (children 2 6 7))
(MAIN::node (id 2) (name "Obtain key") (parent 1) (connector OR) (cost 0) (time 0) (exploited FALSE) (probability 0.0) (difficulty 0) (countermeasure FALSE) (children 3 4 5))
(MAIN::node (id 3) (name "Bribe employee") (parent 2) (connector nil) (cost 0) (time 0) (exploited FALSE) (probability 0.0) (difficulty 0) (countermeasure FALSE) (children nil))

```

Fig. 4.4. Example of a converted attack tree to ES facts

- Exception handler is a “try-except” code block that will terminate the processes and return an error message to the user in the case of an error.
- Process termination terminates the process if any error occurs.
- Output Data. Converted rules are exported to the CLP and DAT files (or one combined file if the user selects this option), supported by the JESS ES platform.
- JESS facts. The result of the transformation process is the rules and facts of a JESS-based expert system’s knowledge base.
- Working memory. Automatically generated rules and facts of an ES knowledge base in the CLP format are imported into ES working memory.
- Prepared knowledge base structure. ES should be able to work with the given (imported) facts.

Our method utilizes the prepared knowledge base structure (Fig. 4.5), which accepts facts generated by ATES.

```
(deftemplate node "Data about nodes"
  (slot id
    (type INTEGER))
  (slot name
    (type STRING)
    (default ""))
  (slot parent
    (type STRING)
    (default ""))
  (slot connector
    (type STRING)
    (default nil)
    (allowed-values nil AND OR XOR SAND))
  (slot cost
    (type INTEGER)
    (default 0))
  (slot time
    (type INTEGER)
    (default 0))
  (slot difficulty
    (type INTEGER)
    (default 0))
  (slot probability
    (type FLOAT)
    (default 0.0))
  (slot exploited
    (type SYMBOL)
    (default FALSE)
    (allowed-values TRUE FALSE))
  (slot countermeasure
    (type SYMBOL)
    (default FALSE)
    (allowed-values TRUE FALSE))
  (multislot children
    (type INTEGER)
    (default nil)))
```

Fig. 4.5. Example of the prepared knowledge base structure in JESS

- Inference engine – the JESS inference engine is used, which utilizes the RETE algorithm.
- Perform IT security risk analysis. At this moment, the ES is prepared for IT security risk analysis, utilizing automatically generated knowledge base rules from the attack trees.
- Risk analysis results and recommendations. The expert system, interacting with the user, performs risk analysis and gives results.

While performing the conversion of UATS into the ES knowledge base facts, each attack tree node is transformed into one fact, as presented in Fig. 4.6.

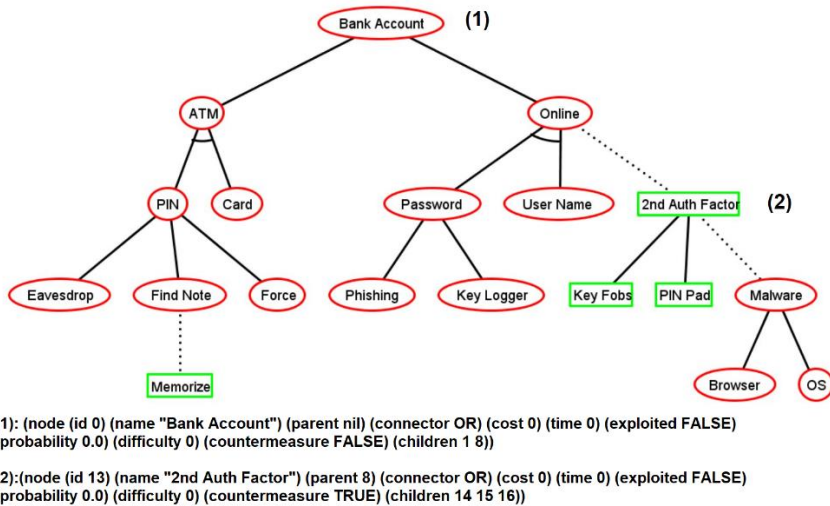


Fig. 4.6. Example of the attack tree transformation to ES knowledge base facts

In the top of Fig. 4.6 shows an attack tree, and in the bottom – two samples of generated ES facts. Every generated fact has the following structure:

- ID of the node. Count from top to bottom and from left to right.
- Name of the node is read from the attack tree.
- Parent depicts the node, which is the parent of the current node. The value is the node ID number.
- Connector defines the node type. Possible values are OR, AND.
- Cost of the attack if this value is specified in the attack tree.
- Time needed to perform the attack if this value is given in the attack tree.

- Exploited state of the node. By default, all values have the FALSE value, and they can be changed during the risk analysis process in the ES. The value can become TRUE based on the user's answers.
- Likelihood of the attack if this value is given in the attack tree.
- Difficulty of the attack if this value is given in the attack tree.
- Countermeasure identifies if the node already has countermeasures applied or not. Possible values are TRUE, FALSE.
- Children show what children this current node has. The value is node ID numbers.

4.2. Proposed Method Implementation

The proposed method was practically implemented. For the method implementation, the ATES tool, which converts AT to ES knowledge base rules, was developed. Fig. 4.7 provides details on the path that attack tree passes in the ATES program.

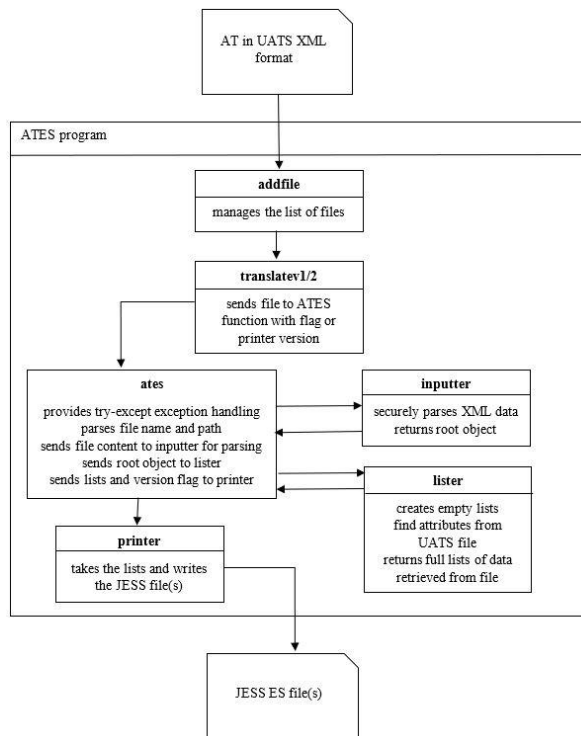


Fig. 4.7. Conceptual Model of ATES

The ATES was written in Python version 3.8 and should be compatible with any Python version greater than 3. Below is provided a short description of functions implemented in ATES:

- Preparing Attack Tree Data (data pre-processing). The creation of an attack tree in one of the various formats accepted by ATTop or direct creation of a UATS XML file must be done by the user. Once the UATS file is returned from ATTop, the ATES program is ready to accept the file for translation.
- Add file. Once the UATS file is returned from ATTop, the ATES program is ready to accept the file for translation. ATES has a simple GUI created using the Tkinter package in Python.
- Translatev1/2 function is wholly encased in a “try-except” block which is a built-in function of Python which is designed for error handling. Any errors from any of the functions within this block result in the deletion of the data, and because of how this “try-except” block is configured, an error message gets displayed on the screen.
- Ates. The *ates* function first takes the files being in input and retrieves the filename, then sends the file to the *inputter* function for parsing, then sends the parsed data to the *inputter* function to extract the necessary node information into lists, before finally sending the lists to the printer function which will write the files. The *ates* function will also take the value sent with file data from the translate functions, called *pvalue*, and will modify the new file name to differentiate between version one and version two if the *pvalue* is 2.
- Inputter function parses the XML file using the defused XML ElementTree package built into Python. Using the defused XML package prevents code execution from maliciously crafted XML files. The function returns an XML Document Object Model, or DOM, for use by the lister function.
- Lister function takes the XML DOM from the inputter function and extracts the necessary data from it. It does this by first creating empty lists, then finding the attributes to fill each list. It then returns the full lists to the *ates* function to be sent to the printer function.
- Printer function has a “try-except” block for creating a directory to store the output files from the conversion process. The files created by this function will also expect this directory to exist and will provide the user with the command to run if the files are in this directory. Next, the function will open the new *.clf* file in write mode or create it in write mode if it does not exist and write the JESS program code which has been written. If *pvalue* is 1 then the function will write the necessary JESS code and

the list data from lister into the same file, but, if *pvalue* is 2, then it will write an additional JESS code and create a new *.dat* file for containing the attack tree's node data. All the files will be named automatically based on the original input file name. Once these processes have been completed, the files will be saved and closed.

Successful attack tree conversion to JESS expert system knowledge base is demonstrated in Fig. 4.8. Converted files are automatically exported to the directory "Attack_trees/JESS" in CLP format.

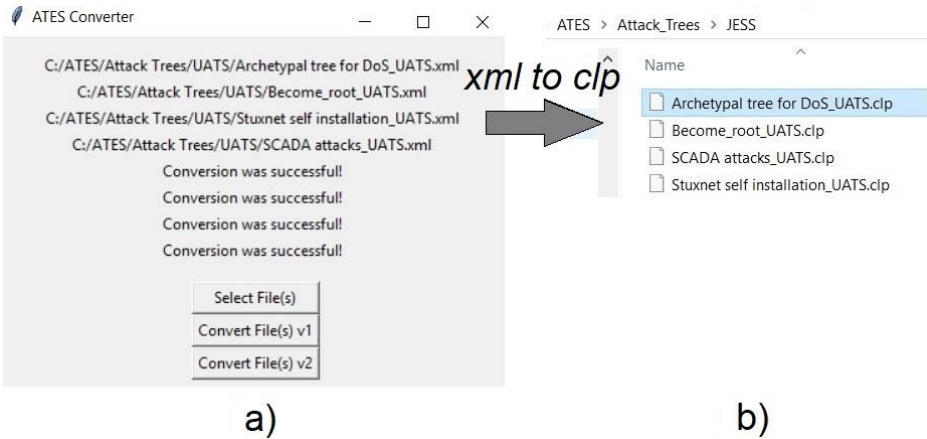


Fig. 4.8. Example of successful conversion to the JESS expert system: a) successfully converted xml files; b) automatically exported clp knowledge base files

- JESS ES file(s). The produced JESS programs have an initial welcome message at the top of each file with instructions on where to place the files and with what JESS command to run them. Successful JESS Operation is shown in Annex B Fig. B.3.

4.3. Proposed Method Experiments and Results

Several experiments using the proposed method were performed to demonstrate its reliability and applicability. Automatically generated rules and facts of the ES knowledge base were imported into the created ES (based on JESS), which allows performing risk analysis using the generated knowledge base.

The attack trees used for testing ATES' accuracy and capability for error handling have been sourced through numerous scientific articles covering various topics related to attack trees, attack tree examples from attack tree tools, and from

results in Internet searches for attack tree examples. Additionally, the Mitre organization has a plethora of information that is easily turned into example attack trees, such as what is available within the CAPEC¹ and Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK)² databases. The citations for the use of these attack trees have been listed beneath the root node of the attack tree as a comment within the pre-ATTop conversion XML file for that attack tree. The applicability of the method proposed for the automated formation of an expert system's knowledge base was evaluated and approved experimentally.

Table 4.1 indicates the number of attack trees used in this process and their top-level sources. The uniqueness of the proposed approach is the provided possibility to generate an integral knowledge base of an expert system by integrating different attack trees taken from various sources.

Table 4.1. Attack Tree Sources

Attack Tree Sources	Quantity of Attack Trees
A safety/security risk analysis approach of Industrial Control Systems: A cyber bowtie – combining new version of attack tree with bowtie analysis (Abdo et al., 2018)	7
Attack-Defense Tree Tool (ADTool) example attack trees (<i>Library of Attack Trees</i> , n.d.)	10
Attack Modelling for Information Security and Survivability (Moore et al., 2001)	2
Attack tree-based evaluation of physical protection systems vulnerability (Vintr et al., 2012)	1
Attack trees with sequential conjunction [42]	2
Automated Generation of Attack Trees by Unfolding Graph Transformation Systems (Huistra, 2016)	2
Beyond attack trees: dynamic security modeling with Boolean logic Driven Markov Processes (BDMP) (Pietre-Cambacedes & Bouissou, 2010)	2
Bridging two worlds: reconciling practical risk assessment methodologies with the theory of attack trees (Gadyatskaya, Harpes, et al., 2016)	1
Crowdsourcing Computer Security Attack Trees (Tentilucci et al., 2015)	1
DAG-based attack and defense modeling: don't miss the forest for the attack trees (Kordy et al., 2014)	1
Effective Analysis of Attack Trees: A Model-Driven Approach (R. Kumar et al., 2018)	1

¹ <https://capec.mitre.org/>

² <https://attack.mitre.org/>

End of Table 4.1

Attack Tree Sources	Quantity of Attack Trees
Is my attack tree correct? (Audinot et al., 2017)	1
Mission-oriented risk and design analysis of critical information systems (Buckshaw et al., 2005)	1
Multi-vendor Penetration Testing in the Advanced Metering Infrastructure (McLaughlin et al., 2010)	3
Producing and Evaluating Crowdsourced Computer Security Attack Trees (Bogaard et al., 2016)	1
Risk assessment of cyber-attacks in ECPS based on attack tree and AHP (Ru et al., 2016)	1
SANS ICS Attack Surfaces (Bellovin, 2016)	1
Security risk assessment framework for a smart car using the attack tree analysis (Kong et al., 2018)	5
Semi-automatically Augmenting Attack Trees using an Annotated Attack Tree Library (Jhawar et al., 2018)	2
Studying Cyber Security Threats to Web Platforms Using Attack Tree Diagrams (Petrica et al., 2017)	1
Survivability analysis of SOA based on attack tree models (Y. Xiao et al., 2012)	1
Understanding risk through attack tree analysis (Ingoldsby, 2004)	2
The total quantity of sourced attack trees:	49

In the experiment, 22 different free sources with attack trees were analyzed, and 49 different attack trees were transformed. One attack tree, in general, is dedicated to one main attack, so it allows ES to assess the risk for 49 different types of attacks. During the experiment, a ruleset composed of 875 facts was created. These facts allow assessing the risk of 49 attacks and analyzing 826 controls (Fig. 4.9). This can be considered a significant achievement in information security risk management since effective and source-independent utilization of existing and reliable information sources and their adaption for use in expert systems creates preconditions for performing information security risk analysis even by non-security specialists.

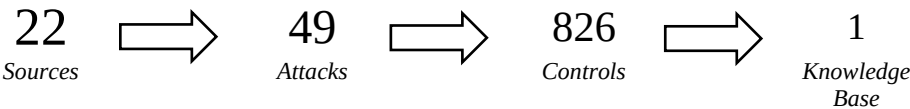


Fig. 4.9. Summary of Transformation Results

The experiment had shown that out of approximately 50 attack trees, there were zero errors with the ATEs conversion program when inputs were correctly configured. When a non-XML file is submitted, a syntax error is raised, and the data is destroyed by the program. If, however, a file is in XML but does not contain the expected formatting, then a syntax error is displayed, and a file is created with pythonic “none” data instead of the attack tree node data. Additionally, the resulting JESS expert systems were functioning as intended.

To demonstrate that automatically generated rules are practically applicable, the JESS-based expert system prototype was used. The generated rules, including information attack success rates, were imported from the tree, presented in Fig. 4.10.

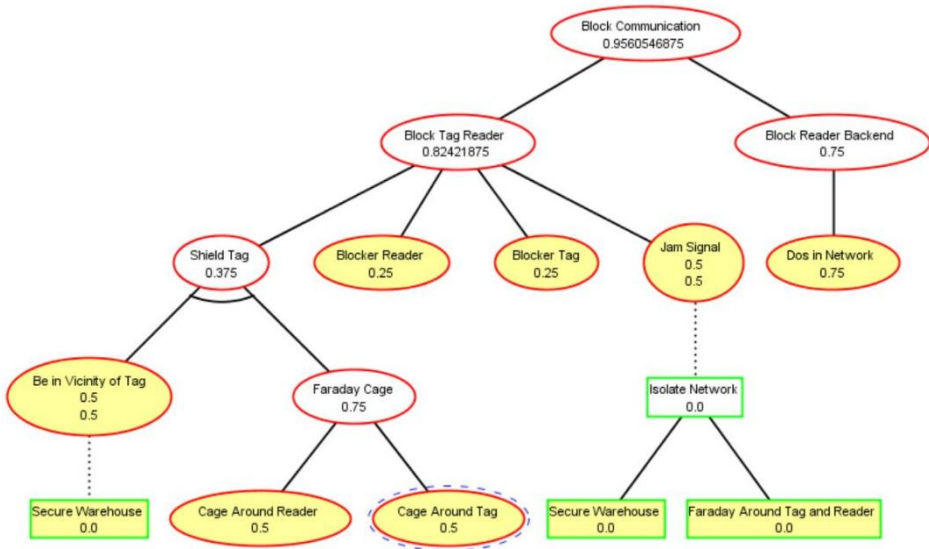


Fig. 4.10. Attack tree “How to block the communication between an RFID reader and tag?”

This example utilizes the “How to block the communication between an RFID reader and tag?” attack tree. In each tree, information on attack likelihood success is stored. Nodes marked red are attack representing nodes, while green nodes define countermeasures, i.e., defense nodes. As stated earlier, this attack tree, stored in XML format, was converted using the proposed method and later imported into the existing expert system, used for risk analysis, prototype. Impact estimation at the current step was obtained from the user input.

Risk calculation was made by the classical equation:

$$RISK = LIKELIHOOD \times IMPACT.$$

(4.22)

The sample of the obtained risk evaluation is presented in Fig. 4.11.

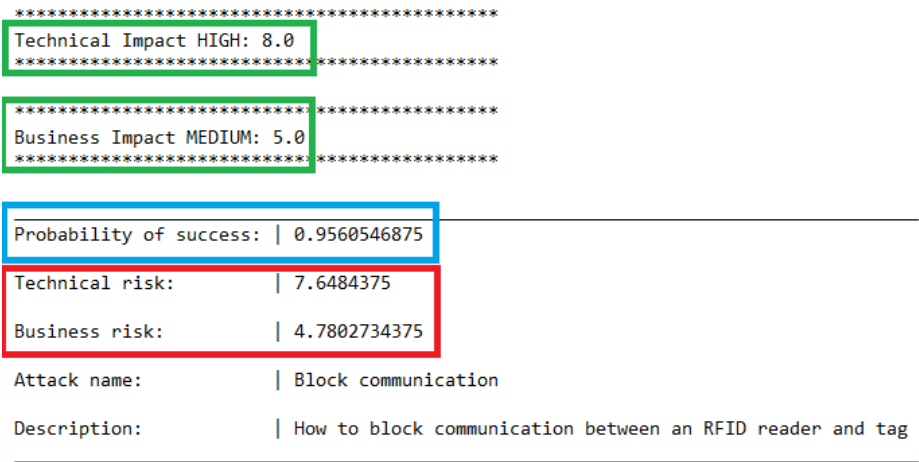


Fig. 4.11. Example of the risk calculation

The green color marks the impact entered by the expert system user. It can also be imported from the attack tree if such information is included in it. The blue color denotes the likelihood that it is imported from the attack tree. The likelihood can vary depending on the availability or absence of countermeasures. In this sample, the likelihood is calculated for the case when no countermeasures are applied. The red color is used to highlight the risk, calculated by the expert system according to the equation provided above. Since the ES, evaluating both technical and business risks was utilized for the experiments, they both can be seen on the screen.

The provided example should not be seen as finite or unconditional. It is just used to demonstrate that the rules generated by the method proposed can be integrated and used in the new or existing expert system dedicated to information security risk analysis. The knowledge base created can be utilized not only for the risk calculation but also for evaluating other criteria and estimations, which are important in the risk assessment process. The method proposed can transform and import any information stored in a tree as likelihood, price of the attack, weight coefficients, impact, and others. Even more, information imported can be used for detecting the most probable attack vector or for making suggestions on possible controls and countermeasures. Still, this is already a part of the specific expert

system decision-making process, which is out of the scope of this research and can be considered a perspective research topic.

As with any methodology, there are some limitations and issues. Below, a short description of such issues is provided with comments on the controls that were or could be applied to minimize their influence on the quality of results:

- One potential issue of the proposed method is that there is a small possibility that data is altered during the transformation processes. To resolve this issue, the data was visually compared during the heuristic phase from the input to the resulting files to verify the accuracy.
- Another weakness of the proposed method is that the output data will only be as reliable as the input data. I.e., if an inaccurate attack tree is converted to the expert system rules, then the expert system will not provide accurate information regarding any eventual risk assessments. Ultimately, no way exists around this issue beyond double-checking that the attack trees are accurate, coming from reliable sources, and apply to the system in question. This will require that users of this method would have some knowledge of their systems and that they are receiving accurate attack tree information.
- Potentially, the greatest issue in this method lies in the extreme variability within XML. This has been mitigated somewhat by using the output from ATTop to standardize the expected input to ATES. However, the naming conventions of various tools being capable of significant differences for the same attributes (e.g., the attribute identifier “duration” referring to the length of time required for an attack could also be called “time” or even “length” by specific tools) means that the only solution is to code each tool’s naming convention into the lister function.

The main advantages of the proposed method are related to the flexibility in data processing and output:

- Due to the popularity and ease of use of the Python programming language, it is relatively easy to increase the acceptable attack tree formats by determining the attribute identifiers used within that tool’s XML format. Utilizing an “if-then” statement within the lister function would allow for the conversion of these attack trees as well.
- The configuration of version two allows for a relatively small set of changes which would mean a single JESS expert system could read the .dat files for multiple attack trees. Because the data is already in a separate file, it would be possible to select multiple .dat files to minimize consumed storage space on a device. This was not pursued by the authors due to a lack of necessity in modern systems.

- Finally, there are several points where data can be configured by the user, and it will be accepted throughout. It is possible to set the attribute values either before translation in ATTop, after translation from ATTop, and before input to ATES, after output from ATES by manually editing the data files, or inside the JESS program created by ATES. The most efficient and accurate method is to either modify the final JESS .dat files or the XML files after being output from ATTop.

Further research on the methodology improvement will be concentrated on:

- Incorporation of PDF files as a source of attack tree;
- Minimizing the time for the file inputting process by creating the automated input file batch process;
- Resolving the issues related to the variability of attack tree representation and naming formats.

4.4. Conclusions of Chapter 4

1. The method was proposed for converting attack trees into a format usable by JESS-based expert systems for reusing the existing attack tree repositories in facilitating information and IT security risk analysis. The method utilizes several tools: already existing ATTop, which performs the initial conversion of attack tree data into the unified UATS XML format, and the newly developed ATES tool, which performs further transformation into the CLIPS rules supported by the JESS expert system platform, which was chosen for the experiments due to its broad adoption.
2. The developed method transforms attack trees into expert system knowledge bases rules, which store specific knowledge needed for information security risk analysis: the purpose of attack, attack steps and relations between them, countermeasures, and attack parameters: cost, time, likelihood, difficulty. In the experiments, attack likelihood, as a parameter, was used, but the method also allows the transformation of other parameters if they are given in the attack tree.
3. The performed experiments have covered data import from attack trees from 22 different sources and allowed the creation of the knowledge base that could be used to assess the risks of 49 cyberattacks and propose more than 826 possible controls. The experiment had shown that of approximately 50 attack trees, there were zero errors with the ATES conversion program when the inputs were correctly configured. The method has demonstrated reliability when

converting attack trees of a supported type. Throughout the heuristic portion of the experiments, the data conversion correctness was confirmed to be evaluable by JESS and tested on the JESS platform to verify that accidental attack tree data modification will not occur during any phase of the translation process. However, recent trends in an attack tree representation, transformation, and generation show a large amount of variability in attack tree representation and generation. This variability leads to some difficulties in the transformation that could require further research.

Prototype Expert System for Information Security Risk Analysis in Small-Medium Enterprises

In this thesis, we present a model of an expert system dedicated to information security risk analysis for SMEs, which gets a knowledge base automatically from existing information security sources since the creation of a knowledge base by expert interviews is timely and expensive. A knowledge base was generated using automatically created expert systems knowledge bases rules, presented in this work in Chapters 2–4. The generated rules were integrated into the JESS-based prototype risk analysis ES, adopted for SMEs.

The proposed expert system prototype was presented in an international journal (Vitkus et al., 2019) and results were presented at the 11th International Workshop Data Analysis Methods for Software Systems (DAMSS 2019), Druskininkai, Lithuania.

5.1. Expert System's Design for Information Security Risk Analysis

The development of an expert system is dedicated to information security risk analysis for SMEs by getting a knowledge base automatically from existing information security sources. This chapter presents the requirements, architecture, and realization for developing such a system.

5.1.1. Requirements for the Expert System

An expert system is developed using the JESS platform. By choosing the JESS expert system's shell, some information system design stages are not needed: security specifications, user interfaces, data processing and database requirements, implementation and its requirements, user documentation, program architecture, component relations, data flows, hardware, and software requirements.

It is necessary to describe the functional and structural requirements of the system for the developed expert system. The knowledge base, its operation, and the sequence of operation of the program are discussed in the previous sections describing the JESS platform. The user interface of the system in the standard JESS environment is simple and sufficient for the expert system's model.

To validate the methods proposed in this thesis and ensure that they meet end-user requirements, some additional functional requirements for the system were developed:

1. The expert system must include information security topics relevant to the SME.
2. The system gets facts from user answers.
3. The user of the system must have the ability to change facts from the dynamic fact base.
4. The questions asked by the system must be relevant to a specific SME business area.
5. The system results must be shown and monitored with the ability to export them into the XML format.
6. Answers to questions must be limited according to the type of question.
7. The system must give recommendations to the user.
8. Security requirements: data integrity and availability.

It is important to mention that the consumer must be able to indicate the significance of the tolerable risk. The level of tolerance in this system is important not only in the risk assessment but also in the ability to use the expert system as a tool to analyze how the risk values will change in the organization or simply explain whether the risk assessment did not include thresholds.

Requirements for the presentation of system results:

1. The system shall only report values for which the calculated risk value is equal to or greater than the specified tolerance.
2. In the results provided by the system, each value with a risk equal to or exceeding the specified tolerable risk shall be separated.
3. When providing the estimated value risk value, the expert system shall also provide the likelihood value that was used in the risk calculation.
4. The risk value measure shall state the reasons for the calculated risk and the possible risk mitigation recommendations to be used.

5.1.2. Architecture of the Developed Expert System

The expert system is designed in the JESS environment, where mechanisms for rule and fact management are already implemented. The facts are stored in working memory, also known as a fact base, which has already been described in the previous sections. Both the initial facts and the facts obtained after activating the rules are stored in the working memory. To accommodate and quickly access the facts, JESS uses a set of individual indices based on a principle like that used in relational databases. Each fact is stored in the working memory in the order in which it was placed.

The components of the developed expert system are shown in Fig. 5.1.

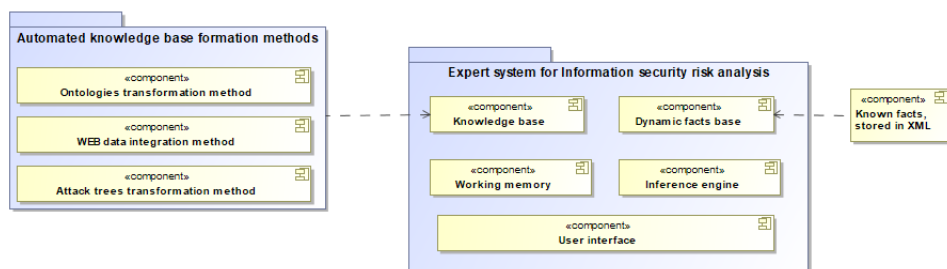


Fig. 5.1. Architecture of the developed expert system

For the expert system developed in this work, the standard structure of the expert system is suitable only partially for several reasons. In addition to the knowledge base, the expert system also uses facts from a dynamic fact base. In this way, if facts from a dynamic fact base are used, they are stored in working memory as higher priority facts, and the system no longer asks questions to find out those facts. If a rule is fired in the system that should change a fact already taken from the fact base, it does not change that fact. Otherwise, if the system does not use facts from an external fact base, all the facts required by the system are known by the user or by shooting certain rules accordingly.

The knowledge base of the expert system in the JESS environment is suitable for the system without additional requirements.

5.1.3. Implementation of the Expert System

The UML use case diagram, given in Fig. 5.2, shows the actors in the system and the use cases related to the actors.

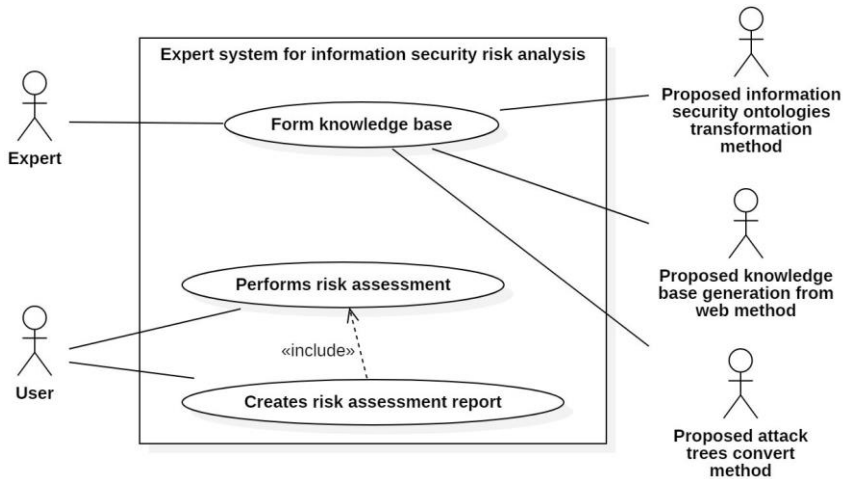


Fig. 5.2. UML use case diagram

Name	Form of a knowledge base
Brief descriptions	Expert system knowledge base is formed using automated knowledge base methods and manually by the expert in the field.
Actors	Expert, user, proposed method for the transformation of information security ontologies, proposed method for the knowledge base generation from the web, proposed attack tree conversion method.
Preconditions	Empty knowledge base is formatted before risk assessment.
Normal flow	Risk assessment can be performed if the knowledge base is relevant.
Alternative flows and exceptions	Knowledge base formation is not needed if the knowledge base is already relevant.

Postconditions	Expert system is ready to use after the knowledge base is formatted.
Name	Perform risk assessment
Brief descriptions	User performs risk assessment using the expert system.
Actors	User
Preconditions	Expert system knowledge base should be formatted or updated if needed.
Normal flow	User answers to questions requested by the expert system.
Alternative flows and exceptions	Risk assessment process can be cancelled.
Postconditions	Risk assessment is performed, ready to create a risk assessment report.
Name	Create a risk assessment report
Brief descriptions	User can create a risk assessment report using expert system results and recommendations.
Actors	User
Preconditions	Risk assessment should be performed using the expert system.
Normal flow	User uses risk assessment results and expert system recommendations.
Alternative flows and exceptions	Risk assessment report can be skipped if not needed.
Postconditions	User gets the risk assessment results with recommendations for needed actions to mitigate the identified risks.

The risk assessment in the expert system is performed according to the process diagram given in Fig. 5.3.

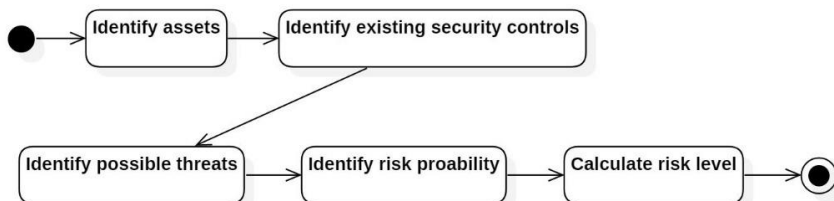


Fig. 5.3. Risk analysis process diagram

Risk analysis in the expert system consists of two main parts: asset identification and risk calculation. The results of the risk analysis provide the basis for risk assessment, risk management, and risk acceptance. When providing results or recommendations, the expert system will present them to the user according to the risk. Control actions and cost-benefit analysis will not be provided as it is out of the scope of this work.

In a quantitative analysis, the level of risk is indicated by a numerical value corresponding to the impact and probabilities. Using quantitative analysis, it is possible to predict the quantitative value of individual risks and the risk of the whole study, which is relevant within the framework of this system. Regarding small businesses, it should be mentioned that a quantitative approach to assessing the individual potential risks can identify the most likely risks that could lead to significant losses for small businesses.

Risk analysis, which plays an important role in information security management, and its management in the expert system will be expressed in quantitative terms. The quantitative risk analysis at the scale of this projected expert system is sufficiently detailed. It is designed to quantify complex and diverse risk scenarios. A four-step approach will be used for quantitative risk analysis.

In the designed expert system, the risk analysis is performed in the following stages:

1. Identifying assets;
2. Identifying threats;
3. Assess the importance of assets that are threatened;
4. Indicate the security measures used to reduce the risk.

The risk analysis of the existing threat, the values of which are unacceptable, provides recommendations for risk management.

To adopt the application of the expert system to companies of different sizes and profiles, it was decided to group the questions and rules. In the JESS expert system, this is done by breaking down the desired questions and rules into modules. First, general questions are asked to identify the enterprise profile. This is followed by questions relevant to the enterprise's field of activity and other questions to identify security risks. Once security risks have been identified, the level of risk is calculated, and recommendations for the management of intolerable risks are provided.

The main facts that determine the questions fired by the system are the enterprise profile, the size of the enterprise, and the security policy. Depending on the facts presented, the rules are activated according to the company profile.

A system sequence diagram is shown in Fig. 5.4.

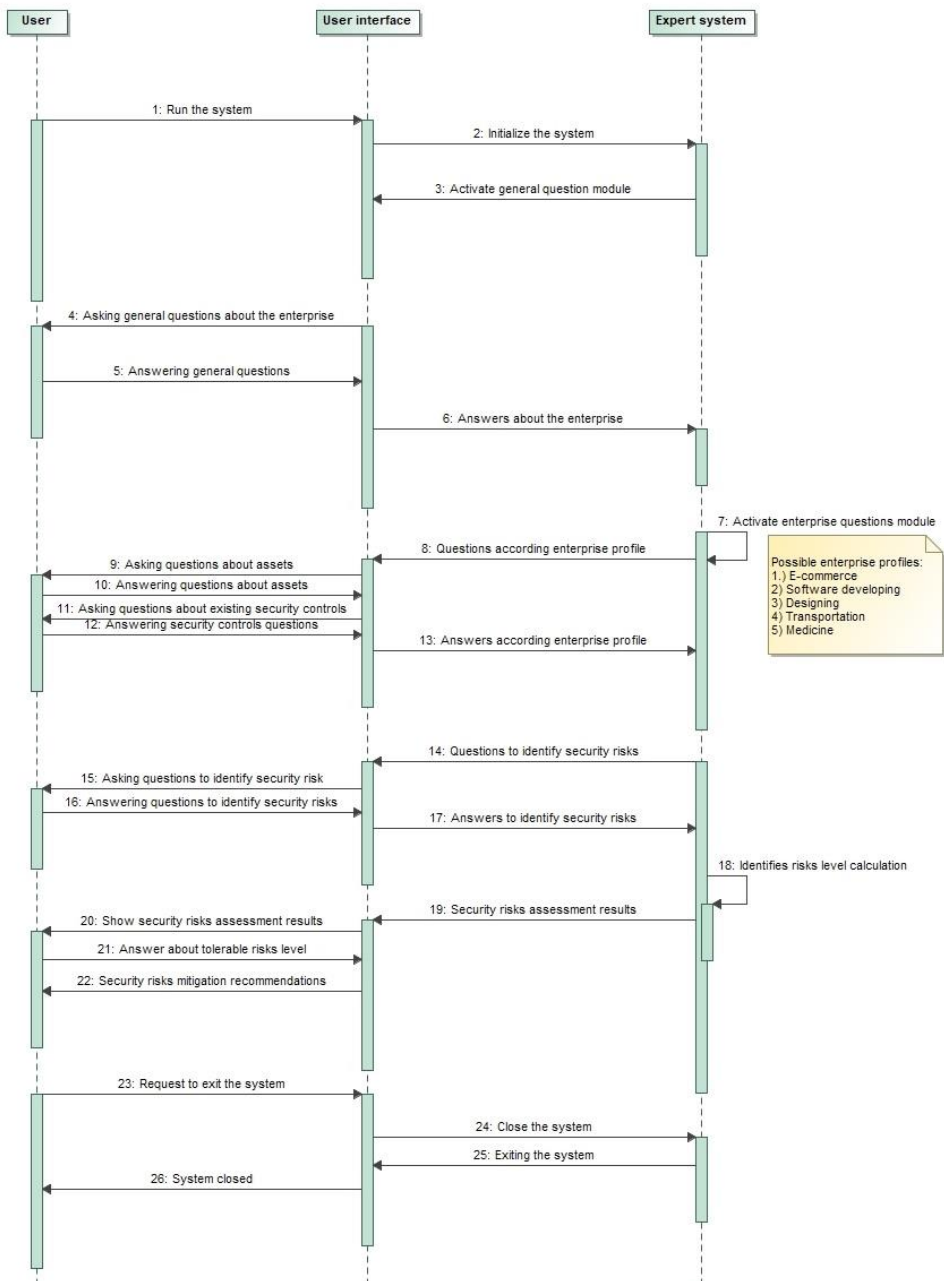


Fig. 5.4. UML Sequence Diagram of the developed expert system

Today almost all companies process data and automate their processes using information technologies. Even the smallest company possess information and information system that must be secured. Information security risk analysis is a compulsory requirement both from the side of regulating documents and the information security management decision-making process. However, it is a challenging task for small and medium-sized enterprises (SMEs) because of the lack of competence and limited resources.

5.2. Developed Expert System

For the validation of the proposed methods, the JESS-based expert system was developed. The system is dedicated to information security risk analysis in SMEs, according to functional requirements given in the previous sub-chapter.

The developed expert system is JESS-based, and the user interacts with its native user interface. In these experiments, a textual interface was used. The developed expert system has five profiles of SMEs:

- 1) E-commerce,
- 2) Software development,
- 3) Designing,
- 4) Transportation,
- 5) Medicine.

Some of the questions in the expert system are dedicated to a specific profile of the organization, and there is no need to answer irrelevant questions. The experiments were performed with two profiles, as attracting experts in the field is a resource-intensive process.

In this thesis, generated rules were integrated into this expert system's knowledge base. This part of the expert system's knowledge base was developed using traditional methods for the development of a knowledge base and included rules for identification of appropriate assets, calculating impact and probabilities based on the environment of a specific company (infrastructure, maturity level, environment, sector, etc.), while rules generated automatically from the ontology, web data and attack trees included additional information on appropriate security controls, malware, and threats.

The JESS platform was used, utilizing the RETE algorithm (rules and facts are changing in different periodicity (Peltier, 2005)), which optimizes rule working cycles and minimizes the probability of errors (E. J. Friedman-Hill, 2007b).

The prototype of an expert system makes use of dynamic facts from the earlier answer. If there are facts in the dynamic facts base, they are loaded into memory with higher priority than knowledge base facts. The rule that can change the known fact is not activated either. Basic questions are oriented toward small

and medium-sized enterprises by their business profile. The risk analysis value is numeric. This risk analysis process includes four main steps: (1) identifying assets, (2) identifying possible threats, (3) business impact, and (4) recommendations on risk reduction.

The result of risk value is the sum of evaluated asset value and likelihood. The likelihood, according to ISO 31000, means the “chance of something happening.” The English term “likelihood” does not have a direct equivalent in some languages; instead, the equivalent of the term “probability” is often used. However, in English, “probability” is often narrowly interpreted as a mathematical term. Therefore, in the risk management terminology, “likelihood” is used with the intent that it should have the same broad interpretation as the term “probability” has in many languages other than English (ISO, 2009). Descriptions of used risk likelihood values are provided in Table 5.1.

Table 5.1. Risk likelihood value descriptions

Category	Value	Description
Very likely	4	Risk likelihood is very high, i.e., equal to or greater than 81 % certainty that it occurs.
Likely	3	Risk likelihood is high, i.e., there is between 61 % to 80 % certainty that it occurs.
Possible	2	Risk likelihood is medium, i.e., there is between 41 % to 60 % certainty that it occurs.
Unlikely	1	Risk likelihood is low, i.e., there is between 21 % to 40 % certainty that it occurs.
Very unlikely	0	Risk likelihood is very low, i.e., there is between 1 % and 20 % certainty that it occurs.

Risk calculation is based on the matrix shown in Table 5.2. The risk value can vary from 0 to 8. A lower value shows a lower risk.

Table 5.2. Used risk calculation matrix

Asset value	0	1	2	3	4
Likelihood					
0	0	1	2	3	4
1	1	2	3	4	5
2	2	3	4	5	6
3	3	4	5	6	7
4	4	5	6	7	8

Expert system questions were adapted to five different SME business profile types: e-commerce, software development, design, transportation, and private medical practice. The main factors that determine questions activated by the system are enterprise size and business profile.

As the authors of other expert systems mention, the broader description of the problem in the rules gives the greater accuracy of the answers, which is not necessarily true in the case of people, as already mentioned. Therefore, such a system requires automated testing, which should theoretically cover almost all possible combinations of rules and facts, and in real terms, cover as many possible variants as possible, as the expert system demonstrates artificial intelligence. Also, by starting to test the developed system in the early stages, it is possible to avoid several levels of mistakes, which are very easy to leave in the expert system and to correct them, it would be necessary to write new conflict-solving rules.

To make sure that the developed expert system works correctly, it was decided to perform its automatic testing, dividing it into two parts: automatic random and automatic target testing. In automated random testing, the performance of the expert system model and its results are tested with random values, and in automated random testing, the expert system is initially provided with the desired facts and only then random responses. Testing in automated targeted testing initially provides the desired facts that can lead to further questions from the expert system and the final results. By testing in this way, it is easy to discover the erroneous rules observed.

In both types of testing, monitoring of facts, rules, and their activation are enabled to verify who and what fired the rule.

For automated random testing, it was decided to use standard Linux CLI tools and integrate them into the bash script. Test logs - reports are saved, and an XML file of facts and rules is provided to JESS for later review.

It is possible to use a Linux seq or Excel spreadsheet to generate an answers file. In the case of expert system testing, the answers file was generated using an Excel spreadsheet containing two types of answers, as the expert system currently accepts only two types of answers: textual answers only “yes” or “no” and numeric values. The use of such a file for testing at the same time checks whether the expert system accepts the answers in the correct form specified in the question.

Testing consists of four parts: shuffling the generated random responses, launching the program, and non-interactively submitting responses from the file in which the responses were shuffled, saving the raw report, processing the report, and saving to the file specified at the beginning of the program. Fig. 5.5 shows the source code for the b.sh script.

```

echo "Iveskite ataskaitos pavadinima"      #ataskaitos pavadinimo prasymas
read atas                                #ataskaitos failo nurodymas
echo "Ataskaitos failas" $atas".txt"      #pilnas ataskaitso pavadinimas
shuf rand.txt >rand1.txt                  #maisomas sugeneruotu atsakymu failas
java.exe -classpath jess.jar jess.Main risk.clp < rand1.txt > ats.txt
sed '1,14d' ats.txt >atas.txt              #programos informacijos trinimas is ataskaitos
sed 's/#/\n&/g' ats.txt >ats.txt           #ataskaitos skaidymas eilutemis
sed -i -e '1d' ats.txt                    #pirmos tuscios eilutes trinimas
pr -tmJ ats.txt rand1.txt > atas.txt       #taisykliu ir atsakymu formatavimas i ataskaita
awk '$2>="$$$$"' atas.txt > "$atas".txt    #ataskaitos irasymas i faila
awk '{a[$4]=$0} END {for (i in a) print a[i]]' "$atas".txt >atas.txt
sed -i -e '$a\' "$atas".txt               #naujos eilutes iterpimas
sed -i.bck '$s/$/~~~~~/' "$atas".txt sed '1!G;h;$!d'
atas.txt >>"$atas".txt
read -p "Testas baigtas"                  #EOB

```

Fig. 5.5. Expert system's automatic testing script

Another way of testing is automatic random testing. It is a very convenient way to test all the rules and the operation of the whole expert system. Such automated random testing is ineffective when supplementing the expert system with new rules or revising them. It was decided to use automatic targeted testing for such tests. Compared to automatic randomization, it differs in that testing initially presents specific facts for testing, and only in their absence is testing continued using random facts. The facts are presented to the system using an XML fact file that is read when the expert system becomes operational or by using rules to record the facts directly into the expert system, which is very inconvenient and increases the likelihood of errors.

An example of automated testing is given in Annex B Fig. B.4. By specifying the necessary facts, each rule can be tested individually or only in part, with further testing using random values. Once the expert system has received the facts, the questions which are intended to obtain those facts shall no longer be asked unless an adjustment rule is issued.

5.3. Experiments and Validation

The expert system's validation with the combined knowledge base (automatically generated and prepared rules) tests included a comparative analysis of results obtained by the expert system and eight independent human experts that got the same company profiles and facts as the expert system. The experts' verification was performed before including them in the experiment. The experts had to perform

tasks that allowed to ensure the sufficiency of their knowledge in information security risk analysis, and only after successful results do experts get a task to perform a risk analysis.

There were 30 questions (15 in each scenario). Questionnaires were constructed in a way to make answering possible for staff with low IT and security knowledge levels. The experimental questionnaire issues included the following threats: IT hardware, physical, software, data leakage, and third-party services. In both scenarios, fourteen questions were the same, only the facts were different. Each scenario had one question tailored exactly to the company profile described in the scenario. All facts used in the design of the questionnaire were initially submitted to the expert system. The main facts used in the experiment are given in Table 5.3.

Table 5.3. Scenarios for risk analysis

Facts	Scenario 1	Scenario 2
Enterprise profile	e-commerce	software developing
Number of employees	5	80
Number of equipment	20	10
Security policy	No	Yes

Experts had to indicate only the risk likelihood by evaluating the provided facts: asset value, deployed controls, available equipment, and other information from the company profile. The final risk value was calculated according to the risk assessment matrix presented in Table 5.2.

The results of the experiment are presented in the form of diagrams below and in more detail. First, experts and expert system results of the risk likelihood evaluation were presented and analyzed, and later, risks values were presented and analyzed. In Fig. 5.6, the results of the risk likelihood evaluation of scenario 1 are given.

For assessing the agreement among the experts, Kendall's coefficient of concordance (also known as Kendall's W) was used. If the test statistic, W is 1, then all the survey respondents have been unanimous, and each respondent has assigned the same order to the list of concerns. If W is 0, then there is no overall trend of agreement among the respondents, and their responses may be regarded as essentially random. Intermediate values of W indicate a greater or lesser degree of unanimity among various responses.

The definition of Kendall's W . Suppose that object i is given the rank r_{ij} by expert number j , where there are in total n objects and m experts. Then the total rank given to object i is

$$R_i = \sum_{j=1}^m r_{i,j}, \quad (5.1)$$

and the mean value of these total ranks is

$$\bar{R} = \frac{1}{n} \sum_{i=1}^n R_i. \quad (5.2)$$

The sum of squared deviations, S , is defined as

$$S = \sum_{i=1}^n (R_i - \bar{R})^2, \quad (5.3)$$

and then Kendall's W is defined as (Dodge et al., 2006):

$$W = \frac{12S}{m^2(n^3 - n)}. \quad (5.4)$$

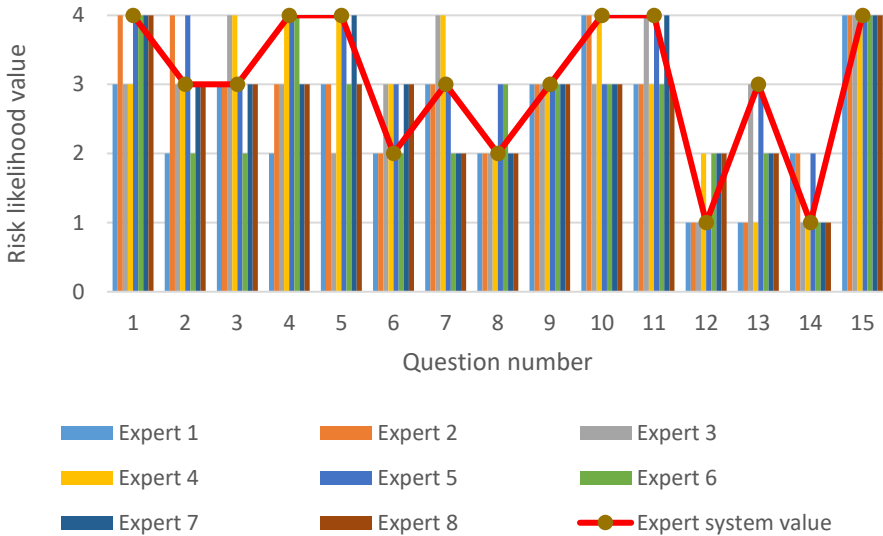


Fig. 5.6. Risk likelihood evaluation results of scenario 1

The graph in Fig. 5.6 demonstrates that expert system values and the average of experts' values are similar. Equation (5.4) $W = 0.573$ indicates a relatively high level of agreement between the experts. Next, the p -value was also evaluated. P -value is a probability that measures the evidence against the null hypothesis. Lower p -values provide stronger evidence against the null hypothesis. Scenario 1 p -value = $2.19\text{E-}08 < 0.05$, thereby allowing us to reject the null hypothesis that there is no agreement among the experts.

In Fig. 5.7, experts' values in scenario 1 are given in a Boxplot diagram, which allows visually identifying questions where experts' opinion match or differs mostly.

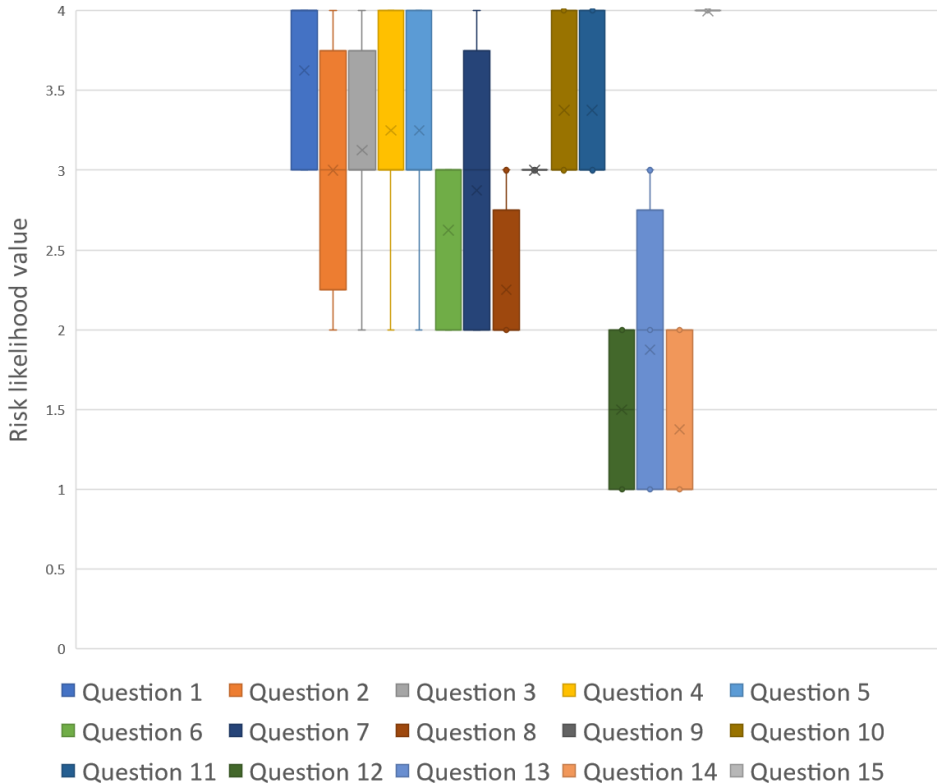


Fig. 5.7. Experts' values in scenario 1

Based on the figure, all expert opinions matched in questions 9 and 15, and they had some differences in questions 7 and 13.

Fig. 5.8 gives the results of the risk likelihood evaluation of scenario 2. This figure shows visually bigger differences between experts and expert system values than in the first scenario, but no question exists where expert system values did not match with at least one expert. According to equation (5.4), $W = 0.236$, which indicates a relatively low level of agreement between the experts, and $p\text{-value} = 0.023 < 0.05$, thereby allowing us to reject the null hypothesis that there was no agreement among the experts. The agreement between the experts in the second scenario was lower than in the first scenario.

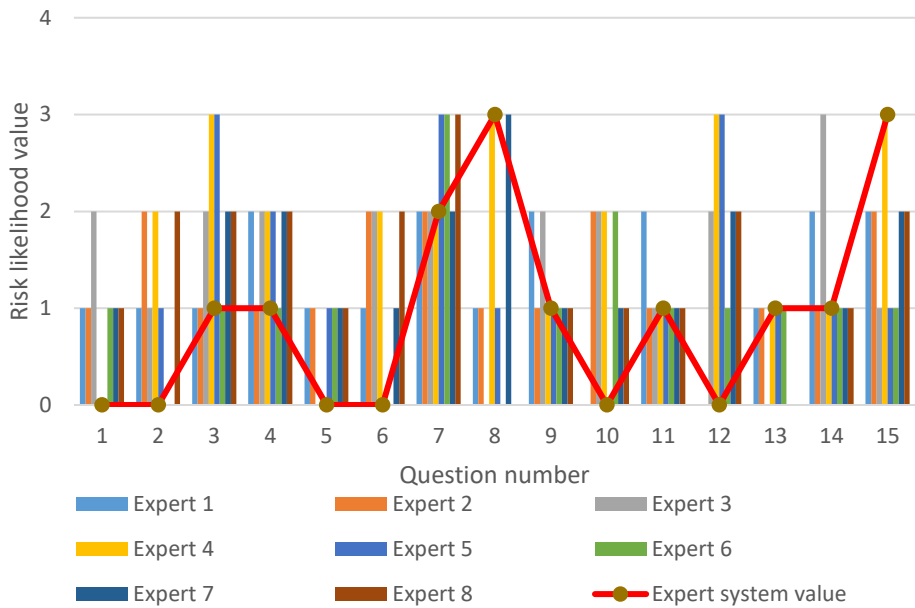


Fig. 5.8. Risk likelihood evaluation results of scenario 2

In Fig. 5.9, expert values in scenario 2 are given in a Boxplot diagram.

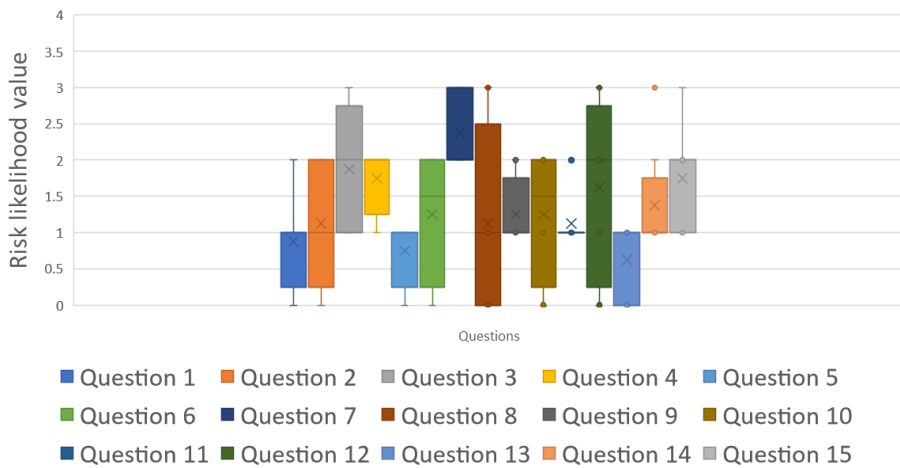


Fig. 5.9. Experts' values in scenario 2

Based on the figure, expert opinions matched in questions 4, 5, and 11, and they had some differences in question 8 and 12.

For the next analysis, an average of expert values was chosen to compare with expert system values. In Fig. 5.10, values given by experts and the expert system for risk likelihood in Scenario 1 are given, and in Fig. 5.11, the same values for Scenario 2 are given.

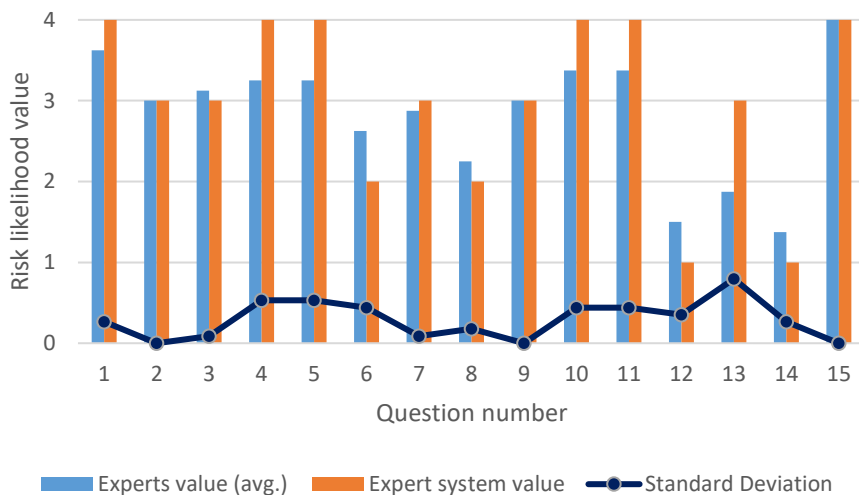


Fig. 5.10. Risk likelihood values given by experts and expert system for scenario 1

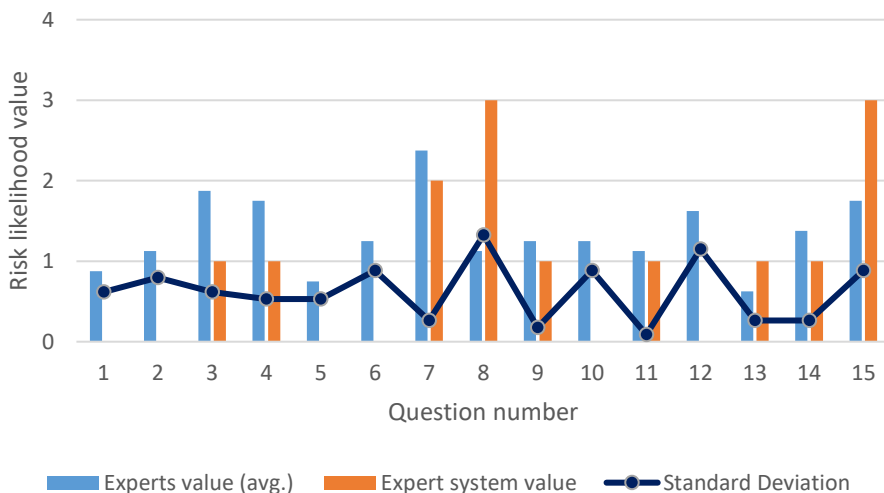


Fig. 5.11. Risk likelihood values given by experts and expert system in scenario 2

The comparative analysis of risk analysis results by the expert system and the average result of human experts' evaluation is provided in Fig. 5.12 (Scenario 1) and in Fig. 5.13 (Scenario 2).

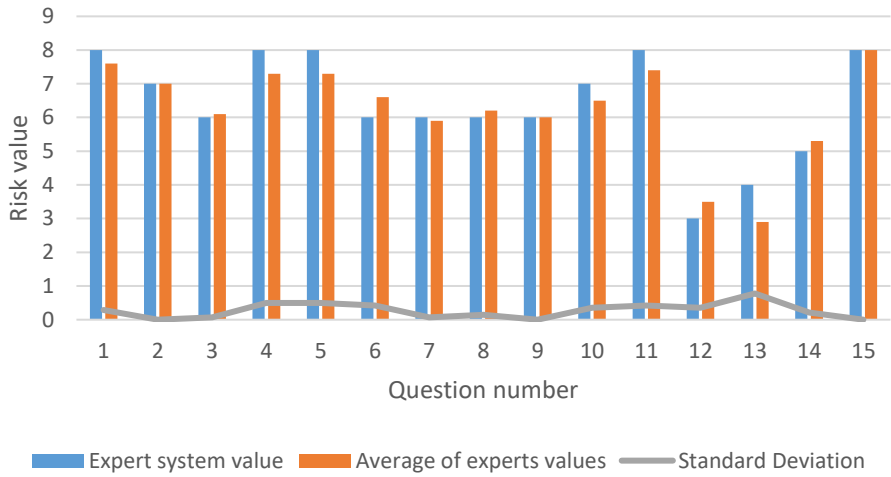


Fig. 5.12. Risk values given by the expert system and real experts for scenario 1

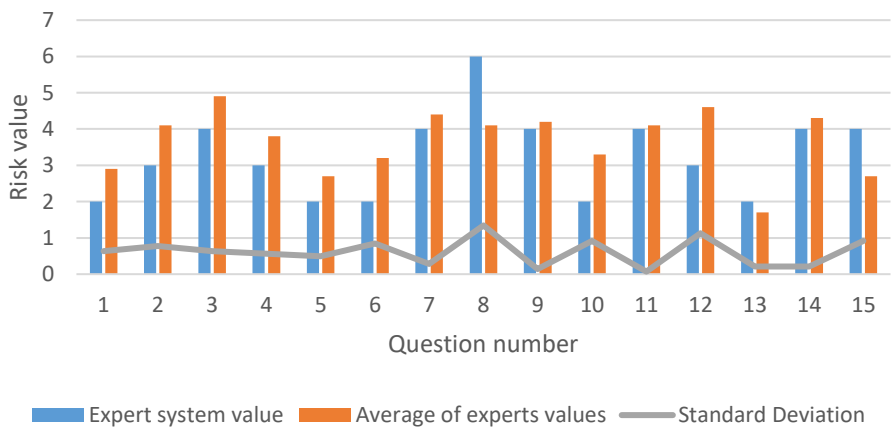


Fig. 5.13. Risk values given by the expert system and real experts for scenario 2

Looking at the likelihood value diagrams, the assessment of the probabilities of the experts is very similar to the assessment provided by the expert system. Still, there are some differences of opinion in the expert assessments. In the case of marginal values, less precision is observed as the identification of risk for a completely insecure or highly protected assets requires additional facts or a more detailed risk assessment, which was not provided in this case.

The risk value diagrams above once again confirm that the risk assessment of the expert system is sufficiently accurate compared to the assessment provided by the experts. Analyzing the likelihood and risk assessment diagrams together, the accuracy of the assessment depends on the assessment values. In the case of marginal values, less precision is observed as the identification of a risk for a completely unsafe or highly protected value requires additional facts or a more detailed risk assessment, which was not provided in this case.

The assessment accuracy was not high in the case of questions and reasons for the expert assessment where the technical equipment or tools are not directly related to the profile of the company presented in the scenario.

The results demonstrate that evaluation given by the expert system correlates in many cases with evaluation by human experts that are taken as a reference.

The correlation coefficient was calculated by the Pearson correlation formula:

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}, \quad (5.1)$$

where r – Pearson correlation coefficient, x is the expert system risks values, and y – experts risk values. If the resultant value – r is close to +1, this indicates a strong positive correlation. If the resultant value – r is close to -1, this indicates a strong negative correlation.

In the first scenario, the correlation coefficient is 0.949, and the p-value is 3.36E-07, while in the second scenario – 0.588, and the p-value 0.039. Correlation coefficients of both scenarios allow us to confirm that experts and expert system results correlate (in the first scenario, strong correlation, and in the second, moderate correlation). P-values of both scenarios are less than 0.05 that allowing us to reject the null hypothesis that there is no agreement among the experts and expert system.

But it is even more important to state that the expert system was able to process the generated rules from the information security ontology, web data, and attack trees, thus proving the concept of automatic rule generation utilizing methods proposed in this work. As the experiments show, the information security risk analysis of SMEs can be performed by one person using the proposed methods and expert system. The classical risk analysis process requires a group of several

people and experts. Thus, it can be confirmed that the proposed and implemented methods will allow small and medium-sized enterprises to perform information security risk analysis with fewer human resources.

5.4. Conclusions of Chapter 5

1. A developed expert system operates knowledge in three ways: from the knowledge base, from the dynamic facts base, or from both. The possibility of using the dynamic facts base allows reusing historical data if the facts remain unchanged. The expert system knowledge base was created in four ways: by translating information security ontologies, transforming malware information from web data, transforming attack trees, and manually creating rules. Thus, the successful use of such a knowledge base show that it is possible to integrate rules obtained in different ways into a single knowledge base.
2. A prototype of a JESS-based expert system dedicated to information security risk analysis in SMEs was developed, where the knowledge base was developed and updated by the methods proposed in this thesis with automatically generated rules. Performed experiments validated the proposed methods. Rules generated using these methods were successfully used in practice. The developed expert system was adopted in five different profiles of enterprises, and two of them were tested with real experts.
3. Experiments validated that the results of the expert system of the results and real experts' results were similar. The developed expert system knowledge base was formed using three automated methods presented in this work. Using these methods, 662 rules were imported from ontologies in the expert system knowledge base, 201 from web data, and 826 from attack trees. The experiments showed that the expert system correlates with the evaluation of a real expert with a 0.949 correlation coefficient in the first scenario and 0.588 in the second.

General Conclusions

1. The performed scientific literature review has demonstrated the importance of automated methods for information security risk analysis expert systems knowledge base generation, since classical manual methods of knowledge base generation are too slow and expensive. The current scientific research concentrates on the integration of static data, such as ontologies, but this cannot depict the changing state of the cyberspace risk landscape. The performed analysis has allowed clarifying that research on the improvement of knowledge base generation should be concentrated on the transformation of existing data sources. Three different sources were selected according to the set criteria, such as suitability for information security, adaptability to risk assessment, a sufficient number of information sources, availability in text format, and suitability for automatic transformation: ontology, web data, and attack trees. After the analysis of expert systems platforms, according to the defined criteria, the JESS expert systems shell was selected, which is suitable and sufficient to implement the tasks set in this thesis.
2. This thesis proposed and implemented three knowledge base generation methods using various information sources for automated expert system knowledge base generation: the method based on the transformation of information security ontologies into expert system knowledge base rules and com-

plementing the knowledge base with descriptions of information security controls; method based on the transformation of web data containing malicious code descriptions into expert system knowledge base rules and complementing the knowledge base with dynamic information on the prevalence of malicious code by regions; and method based on the transformation of attack trees from various sources into expert system knowledge base rules that include information relevant to information security risk assessment, such as threat likelihood, cost, complexity, threat exploitation time, and countermeasures. The proposed methods increase the knowledge base creation speed and allow reusing the existing information sources.

3. The proposed methods were verified by ensuring that the automatically generated rules meet the requirements of the syntax and logic of the rules and validated by practical experiments in a developed expert system prototype.
 - 3.1. During the method experiments and validations, two practical information security ontologies were transformed into expert system knowledge base rules with 60% and 64% success rates. In total, 662 rules from ontologies were automatically generated. During the knowledge base formation method based on the acquisition of web data experiments, 201 entries of the knowledge base were formed automatically, where data on geographical malware distribution were collected for three countries: Lithuania, Latvia, and Estonia. Experiments performed on the method of converting attack trees have covered data import from attack trees from 22 different sources and allowed the creation of the knowledge base that could be used to assess the risks of 49 cyberattacks and propose over 826 possible controls.
 - 3.2. The expert system knowledge base was created by combining outputs of the proposed automated methods and the manual expert input. The successful integration of automatically obtained rules into a knowledge base demonstrates the validity of the proposed methods. A further comparative risk assessment by the expert system and human experts has demonstrated the practical applicability for risk analysis in small and medium-sized enterprises since the results of the expert system of the results and real experts' results demonstrate high level of similarity: the correlation coefficients in two scenarios were equal to 0.949 and 0.588, and p-values were less than 0.05 and equal to 3.36E-07 and 0.039, respectively, thus allowing us to confirm agreement among experts and expert system.

References

- Aamodt, A., & Plaza, E. (1994). Foundational Issues, Methodological Variations, and System Approaches. *Artificial Intelligence Communications*, 7(1), 39–59.
- Abdo, H., Kaouk, M., Flaus, J. M., & Masse, F. (2018). A safety/security risk analysis approach of Industrial Control Systems: A cyber bowtie – combining new version of attack tree with bowtie analysis. *Computers and Security*, 72, 175–195. <https://doi.org/10.1016/j.cose.2017.09.004>
- Abdullah, A. A., Fadil, N. S., & Khairunizam, W. (2018). Development of Fuzzy Expert System for Diagnosis of Diabetes. *2018 International Conference on Computational Approach in Smart Systems Design and Applications, ICASSDA 2018*, 1–8. <https://doi.org/10.1109/ICASSDA.2018.8477635>
- Abraham, A. (2005). Rule-Based expert systems. *Handbook of Measuring System Design*.
- Agrawal, V. (2017). A Comparative Study on Information Security Risk Analysis Methods. *Journal of Computers*, 12(1), 57–67. <https://doi.org/10.17706/jcp.12.1.57-67>
- Akerkar, R., & Sajja, P. (2009). *Knowledge-based systems*. Jones & Bartlett Publishers.
- Albakri, S. H., Shanmugam, B., Samy, G. N., Idris, N. B., & Ahmed, A. (2014). Security risk assessment framework for cloud computing environments. *Security and Communication Networks*, 7(11), 2114–2124. <https://doi.org/10.1002/sec.923>
- Alcántara, M., & Melgar, A. (2016). Risk Management in Information Security: A Systematic Review. *Journal of Advances in Information Technology*, 7(1), 1–7. <https://doi.org/10.12720/jait.7.1.1-7>

Alder, H., Michel, B. A., Marx, C., Tamborrini, G., Langenegger, T., Bruehlmann, P., Steurer, J., & Wildi, L. M. (2014). Computer-based diagnostic expert systems in rheumatology: Where do we stand in 2014? *International Journal of Rheumatology*, 2014. <https://doi.org/10.1155/2014/672714>

ATTop. University of Twente - Formal Methods and Tools. <https://github.com/utwente-fmt/attop>

Audinot, M., Pinchinat, S., & Kordy, B. (2017). Is my attack tree correct? *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10492 LNCS, 83–102. https://doi.org/10.1007/978-3-319-66402-6_7

Babu, K. V. S. . J. (2012). Key Factors in Knowledge Management. *SSRN Electronic Journal*, 11(1), 10–24. <https://doi.org/10.2139/ssrn.2150567>

Batista, L., Silva, G., Araújo, V., Araújo, V., Rezende, T., Guimarães, A., & Souza, P. (2018). Fuzzy neural networks to create an expert system for detecting attacks by SQL Injection. *The International Journal of Forensic Computer Science*, 13(1), 8–21. <https://doi.org/10.5769/j201801001>

Bekezhanova, A., & Atymtayeva, L. (2011). Development of Database for Expert Systems. *Computer Modelling and New Technologies*, 15(4), 52–57.

Bellovin, S. M. (2016). *Attack Surfaces*. IEEE Security and Privacy. <https://doi.org/10.1109/MSP.2016.55>

Bennett, S. P., & Kailay, M. P. (1992). An application of qualitative risk analysis to computer security for the commercial sector. *Proceedings - Annual Computer Security Applications, Part F1462*, 64–73. <https://doi.org/10.1109/CSAC.1992.228232>

Berredjem, T., & Benidir, M. (2018). Bearing faults diagnosis using fuzzy expert system relying on an Improved Range Overlaps and Similarity method. *Expert Systems with Applications*, 108, 134–142. <https://doi.org/10.1016/j.eswa.2018.04.025>

Berstel, B. (2002). Extending the RETE algorithm for event management. *Proceedings of the International Workshop on Temporal Representation and Reasoning, 2002*, 49–51. <https://doi.org/10.1109/TIME.2002.1027472>

Blanco, C., Lasheras, J., Valencia-García, R., Fernández-Medina, E., Toval, A., & Piattini, M. (2008). A systematic review and comparison of security ontologies. *ARES 2008 - 3rd International Conference on Availability, Security, and Reliability, Proceedings*, 813–820. <https://doi.org/10.1109/ARES.2008.33>

Bobek, S. (2011). HeaRT rule inference engine in intelligent systems. *Pomiarowy Automatyka Robotyka*, 15, 226–228.

Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., & Smith, M. (2012). OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (Second Edition). *Online*, 27(65), 1–133.

- Bogaard, D., Goel, S., Kandari, S., Johnson, D., Markowsky, G., & Stackpole, B. (2016). Producing and evaluating crowdsourced computer security attack trees. *2016 IEEE Symposium on Technologies for Homeland Security, HST 2016*, 1–4. <https://doi.org/10.1109/THS.2016.7568951>
- Boose, J. H. (1989). A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition*, 1(1), 3–37. [https://doi.org/10.1016/S1042-8143\(89\)80003-2](https://doi.org/10.1016/S1042-8143(89)80003-2)
- Bova, V. V., Kureichik, V. V., & Lezhebokov, A. A. (2014). The integrated model of representation of problem-oriented knowledge in information systems. *8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014 - Conference Proceedings*, 1–4. <https://doi.org/10.1109/ICAICT.2014.7035923>
- Brank, J., Grobelnik, M., & Mladenic, D. (2005). A survey of ontology evaluation techniques. In *Proceedings of the Conference on Data Mining and Data Warehouses*, 166–170.
- Broniukaitis, R. (1997). *Ekspertinės sistemos ir žinių bazės: mokomoji priemonė*. Vilniaus universitetinė leidykla.
- Buckshaw, D. L., Parnell, G. S., Unkenholz, W. L., Parks, D. L., Wallner, J. M., & Saydjari, O. S. (2005). Mission Oriented risk and Design Analysis of critical information systems. *Military Operations Research*, 10(2), 19–38. <https://doi.org/10.5711/morj.10.2.19>
- Burge, J. E. (2001). Knowledge Elicitation Tool Classification. *Artificial Intelligence Research Group Worcester Polytechnic Institute*, 1–28.
- Caldas, M. P. (2003). Management information systems: managing the digital firm. In *Revista de Administração Contemporânea*, Vol. 7, Issue 1, pp. 223–223. Pearson. <https://doi.org/10.1590/s1415-65552003000100014>
- Cao, S. T., Nguyen, L. A., & Szałas, A. (2011). On the web ontology rule language OWL, 2, RL. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6922 LNAI, 254–264. https://doi.org/10.1007/978-3-642-23935-9_25
- Carter, M., Herbsleb, J. D., Lampert, R., Olson, G. M., Olson, J. S., Rueter, H. H., & Størrosten, M. (1992). Collection and analysis of data about group processes in computer supported meetings. *Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems*, 44. <https://doi.org/10.1145/1125021.1125062>
- Chang, J. F. (2016). Business process management systems: Strategy and implementation. In *Business Process Management Systems: Strategy and Implementation*. CRC Press.
- Chen, C. H., & Rao, Z. (2008). MRM: A matrix representation and mapping approach for knowledge acquisition. *Knowledge-Based Systems*, 21(4), 284–293. <https://doi.org/10.1016/j.knosys.2007.07.002>
- Chergui, W., Zidat, S., & Marir, F. (2020). An approach to the acquisition of tacit knowledge based on an ontological model. *Journal of King Saud University - Computer and Information Sciences*, 32(7), 818–828. <https://doi.org/10.1016/j.jksuci.2018.09.012>

- Chung, K., Boutaba, R., & Hariri, S. (2016). Knowledge based decision support system. In *Information Technology and Management*, Vol. 17, Issue 1, pp. 1–3. Springer. <https://doi.org/10.1007/s10799-015-0251-3>
- Cullen, J., & Bryman, A. (1988). The Knowledge Acquisition Bottleneck: Time for Reassessment? *Expert Systems*, 5(3), 216–225. <https://doi.org/10.1111/j.1468-0394.1988.tb00065.x>
- Damaševičius, R., Štulkys, V., & Toldinas, J. (2008). Domain ontology-based generative component design using feature diagrams and meta-programming techniques. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5292 LNCS, 338–341. https://doi.org/10.1007/978-3-540-88030-1_32
- Damaševičius, R., Venčkauskas, A., Toldinas, J., & Grigaliūnas, Š. (2021). Ensemble-based classification using neural networks and machine learning models for windows pe malware detection. In *Electronics (Switzerland)*, Vol. 10, Issue 4, pp. 1–26. <https://doi.org/10.3390/electronics10040485>
- Daraio, C., Lenzerini, M., Leporelli, C., Naggar, P., Bonaccorsi, A., & Bartolucci, A. (2016). The advantages of an Ontology-Based Data Management approach: openness, interoperability and data quality. *Scientometrics*, 108(1), 441–455. <https://doi.org/10.1007/s11192-016-1913-6>
- DeKeyser, F. G., Bar Cohen, B., & Wagner, N. (2001). Knowledge levels and attitudes of staff nurses in Israel towards complementary and alternative medicine. *Journal of Advanced Nursing*, 36(1), 41–48. <https://doi.org/10.1046/j.1365-2648.2001.01941.x>
- Delen, D., & Sharda, R. (2008). Artificial Neural Networks in Decision Support Systems. In *Handbook on Decision Support Systems* 1, pp. 557–580. Springer. https://doi.org/10.1007/978-3-540-48713-5_26
- DENG, W., ZHAO, H., & HE, H. (2018). Research on Heavy Haul Railway Dispatching System Based on Fuzzy Expert System. *DEStech Transactions on Computer Science and Engineering, CCNT*. <https://doi.org/10.12783/dtcse/ccnt2018/24719>
- Desai, N. (2003). *IDS Correlation of VA Data and IDS Alerts*, Vol. 1708, pp. 1–9.
- Dhaliwal, Jasbir S., & Benbasat, I. (1996). The Use and Effects of Knowledge-based System Explanations: Theoretical Foundations and a Framework for Empirical Evaluation. *Information Systems Research*, 7(3), 342–362. <https://doi.org/10.1287/isre.7.3.342>
- Dhaliwal, J. S., & Benbasat, I. (1990). A framework for the comparative evaluation of knowledge acquisition tools and techniques. *Knowledge Acquisition*, 2(2), 145–166. [https://doi.org/10.1016/S1042-8143\(05\)80009-3](https://doi.org/10.1016/S1042-8143(05)80009-3)
- Dipboye, R. L. (1994). Structured and Unstructured Selection Interview : Beyond the Job-Fit Model. *Personnel and Human Resources Management*, 12, 79–123.
- Dodge, Y., Cox, D., & Commenges, D. (2006). *The Oxford dictionary of statistical terms*. Oxford University Press on Demand.

Duque-Méndez, N. D., Tabares-Morales, V., & González, H. (2018). ESIA expert system for systems audit risk-based. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11238 LNAI, 483–494. https://doi.org/10.1007/978-3-030-03928-8_39

ENISA. (2020). *Inventory of Risk Management / Risk Assessment Methods and Tools* . <https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory>

Ericsson, K. A. (2008). Protocol Analysis. *A Companion to Cognitive Science*, 14, 425–432. <https://doi.org/10.1002/9781405164535.ch33>

Feigenbaum, E. A. (1982). Innovation and Symbol Manipulation in Fifth Generation Computer Systems. *Fifth Generation Computer Systems*, 223–226. <https://doi.org/10.1016/b978-0-444-86440-6.50016-5>

Feldman, R., & Sanger, J. (2007). The text mining handbook: advanced approaches in analyzing unstructured data. In *Choice Reviews Online*, Vol. 44, Issue 10. Cambridge university press. <https://doi.org/10.5860/choice.44-5684>

Fenz, S., & Ekelhart, A. (2009). Formalizing information security knowledge. *Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security, ASIACCS'09*, 183–194. <https://doi.org/10.1145/1533057.1533084>

Fenz, S., Plieschnegger, S., & Hobel, H. (2016). Mapping information security standard ISO 27002 to an ontological structure. *Information and Computer Security*, 24(5), 452–473. <https://doi.org/10.1108/ICS-07-2015-0030>

Fila, B., & Widel, W. (2020). Exploiting attack-defense trees to find an optimal set of countermeasures. *Proceedings - IEEE Computer Security Foundations Symposium, 2020-June*, 395–410. <https://doi.org/10.1109/CSF49147.2020.00035>

Focke, F., Maug, E., & Niessen-Ruenzi, A. (2017). The impact of firm prestige on executive compensation. *Journal of Financial Economics*, 123(2), 313–336. <https://doi.org/10.1016/j.jfineco.2016.09.011>

Friedman-Hill, E. (2007). *JESS*. Sandia National Laboratories. <https://jessrules.com/jess/download.shtml>

Gadyatskaya, O., Harpes, C., Mauw, S., Muller, C., & Muller, S. (2016). Bridging two worlds: Reconciling practical risk assessment methodologies with theory of attack trees. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9987 LNCS(318003), 80–93. https://doi.org/10.1007/978-3-319-46263-9_5

Gadyatskaya, O., Jhawar, R., Kordy, P., Lounis, K., Mauw, S., & Trujillo-Rasua, R. (2016). Attack trees for practical security assessment: Ranking of attack scenarios with ADTool 2.0. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9826 LNCS, 159–162. https://doi.org/10.1007/978-3-319-43425-4_10

- Gao, J. B., Zhang, B. W., Chen, X. H., & Luo, Z. (2013). Ontology-based model of network and computer attacks for security assessment. *Journal of Shanghai Jiaotong University (Science)*, 18(5), 554–562. <https://doi.org/10.1007/s12204-013-1439-5>
- Gerber, N., McDermott, R., Volkamer, M., & Vogt, J. (2016). Understanding information security compliance-why goal setting and rewards might be a bad idea. *Proceedings of the 10th International Symposium on Human Aspects of Information Security and Assurance, HAISA 2016*, 145–155.
- Gero, J. S., & Maher, M. Lou. (2013). Modeling Creativity and Knowledge-Based Creative Design. In *Modeling Creativity and Knowledge-Based Creative Design*. Psychology Press. <https://doi.org/10.4324/9780203772881>
- Griffin, N. L., & Lewis, F. D. (1989). *Rule-based inference engine which is optimal and VLSI implementable*. CiteSeer. <https://doi.org/10.1109/tai.1989.65327>
- Grigaliunas, S., Toldinas, J., & Venckauskas, A. (2017). An ontology-based transformation model for the digital forensics domain. *Elektronika ir elektrotechnika*, 23(3), 78–82. <https://doi.org/10.5755/j01.eie.23.3.18337>
- Grigaliunas, S., Toldinas, J., Venckauskas, A., Morkevicius, N., & Damasevicius, R. (2020). Digital Evidence Object Model for Situation Awareness and Decision Making in Digital Forensics Investigation. *IEEE Intelligent Systems*. <https://doi.org/10.1109/MIS.2020.3020008>
- Group, W. C. W., & June, N. (2010). OWL 2 RL in RIF. *Group, June*, 1–265.
- Gruber, T. (2008). Collective knowledge systems: Where the Social Web meets the Semantic Web. *Web Semantics*, 6(1), 4–13. <https://doi.org/10.1016/j.websem.2007.11.011>
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220. <https://doi.org/10.1006/knac.1993.1008>
- Gupta, A., & Hammond, R. (2005). Information systems security issues and decisions for small businesses: An empirical examination. *Information Management and Computer Security*, 13(4), 297–310. <https://doi.org/10.1108/09685220510614425>
- Hamid Mughal, M. J. (2018). Data mining: Web data mining techniques, tools and algorithms: An overview. *International Journal of Advanced Computer Science and Applications*, 9(6), 208–215. <https://doi.org/10.14569/IJACSA.2018.090630>
- Harfouche, A., Quinio, B., Skandrani, S. R., & Marciniak, R. (2018). A Framework for Artificial Knowledge Creation in Organizations. *ICIS 2017: Transforming Society with Digital Innovation*.
- Harrison, L. H., Saunders, P. J., & Saraceni, peter J. (1993). Artificial intelligence and expert systems for avionics. In *Proceedings of the IEEE/AIAA 12th Digital Avionics Systems Conference* (Vol. 11). CRC press. <https://doi.org/10.1109/dasc.1993.283552>
- Hicks, T. (2016). *Global Nuclear Detection Architecture Tool Improvements*. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

- Hill, E. F. (2003). *Jess in Action: Java Rule-Based Systems*. In *Environment*. Manning Publications Co.
- Hoffman, R. R. (1987). The Problem of Extracting the Knowledge of Experts. *AI-Magazine*, 8(2), 53–67.
- Huistra, D. J. (2016). *Automated Generation of Attack Trees by Unfolding Graph Transformation Systems*. University of Twente.
- Ibrahim, F., Basheer Ali, J., Jaais, F., & Nasir Taib, M. (2001). Expert system for early diagnosis of eye diseases infecting the Malaysian population. *IEEE Region 10 International Conference on Electrical and Electronic Technology*, 1, 430–432. <https://doi.org/10.1109/tencon.2001.949629>
- Ingoldsby, T. R. (2004). Understanding risk through attack tree analysis. *Computer Security Journal*, 20(2), 33–59.
- Ingoldsby, T. R. (2013). Attack Tree Threat Risk Analysis. In *Amenaza Technologies Limited*. <https://www.amenaza.com/downloads/docs/AttackTreeThreatRiskAnalysis.pdf>
- ISO. (2009). ISO31000:2018(E): Risk Management: Principles and Guidelines. In *Iso 31000:2018(E)*, Vol. 2018, p. 24. International Organization for Standardization.
- Ivkin, A. N., & Burlakov, M. E. (2020). Realization of Expert Intrusion Detection System Based on the Results of Datasets and Machine Learning Algorithm Analysis. *Прикаспийский Журнал: Управление и Высокие Технологии*, 2, 100–107.
- Jamshed, S. (2014). Qualitative research method-interviewing and observation. *Journal of Basic and Clinical Pharmacy*, 5(4), 87. <https://doi.org/10.4103/0976-0105.141942>
- Janulevičius, J., & Čenys, A. (2014). Development of a Risk Assessment Model for IT Risk Self-Assessment Expert System for SMEs. *International Journal of Computer and Communication Engineering*, 3(4), 306–309. <https://doi.org/10.7763/ijcce.2014.v3.340>
- Janulevičius, J., Marozas, L., Čenys, A., Goranin, N., & Ramanauskaite, S. (2017). Enterprise architecture modeling based on cloud computing security ontology as a reference model. *2017 Open Conference of Electrical, Electronic and Information Sciences, ES-tream 2017 - Proceedings of the Conference*, 1–6. <https://doi.org/10.1109/eStream.2017.7950320>
- Jhawar, R., Lounis, K., Mauw, S., & Ramírez-Cruz, Y. (2018). Semi-automatically Augmenting Attack Trees Using an Annotated Attack Tree Library. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11091 LNCS, 85–101. https://doi.org/10.1007/978-3-030-01141-3_6
- Johnson, F., & Kumar Gupta, S. (2012). Web Content Mining Techniques: A Survey. *International Journal of Computer Applications*, 47(11), 44–50. <https://doi.org/10.5120/7236-0266>
- Joshi, A., Lal, R., Finin, T., & Joshi, A. (2013). Extracting cybersecurity related linked data from text. *Proceedings - 2013 IEEE 7th International Conference on Semantic Computing, ICSC 2013*, 252–259. <https://doi.org/10.1109/ICSC.2013.50>

- Kailay, M. P., & Jarratt, P. (1995). RAMEX: a prototype expert system for computer security risk analysis and management. *Computers and Security*, 14(5), 449–463. [https://doi.org/10.1016/0167-4048\(95\)00013-X](https://doi.org/10.1016/0167-4048(95)00013-X)
- Kalibiatienė, D. (2009). *Ontology Based Development of Domain Rules*.
- Kaminskas, T., Kulvietis, G., & Keblas, L. (2005). Knowledge management in the design of welding conjugations. *Proceedings of the 4th WSEAS/IASME International Conference on System Science and Simulation in Engineering*, 199–202.
- Kapse, C. P., Kumar, A., Dash, M. K., Zavadskas, E. K., & Luthra, S. (2018). Developing textile entrepreneurial inclination model by integrating experts mining and ISM-MIC-MAC. *International Journal of Production Research*, 56(14), 4709–4728. <https://doi.org/10.1080/00207543.2018.1443523>
- Karabey, B., & Baykal, N. (2013). Attack tree based information security risk assessment method integrating enterprise objectives with vulnerabilities. *International Arab Journal of Information Technology*, 10(3), 297–304.
- Katz, Y., & Grau, B. C. (2005). Representing qualitative spatial information in OWL-DL. *CEUR Workshop Proceedings*, 188.
- Kholief, E. A., Darwish, S. H., & Fors, M. N. (2017). Detection of steel surface defect based on machine learning using deep auto-encoder network. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 218–229.
- Khosravi, M., & Fereidunian, A. (2019). Enhancing Smart Grid Cyber-Security Using A Fuzzy Adaptive Autonomy Expert System. *2019 Smart Grid Conference, SGC 2019*, 1–6. <https://doi.org/10.1109/SGC49328.2019.9056611>
- Kiartzis, S. J., Bakirtzis, A. G., Theocharis, J. B., & Tsagas, G. (2000). Fuzzy expert system for peak load forecasting. Application to the Greek power system. *Proceedings of the Mediterranean Electrotechnical Conference - MELECON*, 3, 1097–1100. <https://doi.org/10.1109/melcon.2000.879726>
- Kim, D., Shin, D., Shin, D., & Kim, Y. H. (2019). Attack Detection Application with Attack Tree for Mobile System using Log Analysis. *Mobile Networks and Applications*, 24(1), 184–192. <https://doi.org/10.1007/s11036-018-1012-4>
- Kim, J., & Courtney, J. F. (1988). A survey of knowledge acquisition techniques and their relevance to managerial problem domains. *Decision Support Systems*, 4(3), 269–284. [https://doi.org/10.1016/0167-9236\(88\)90016-4](https://doi.org/10.1016/0167-9236(88)90016-4)
- Kim, S. K., Lim, S., & Mitchell, R. B. (2008). A method for knowledge modeling with unified modeling language (UML): Building a blueprint for knowledge management. In *Current Issues in Knowledge Management* (pp. 228–242). IGI Global. <https://doi.org/10.4018/978-1-59904-916-8.ch015>
- Kireeva, N., Pozdnyak, I., & Gazizulina, A. (2019a). Filling a Knowledge Base for Expert System in Information Security. In *IOP Conference Series: Materials Science and Engineering*, Vol. 618, Issue 1, pp. 57–76. Springer. <https://doi.org/10.1088/1757-899X/618/1/012085>

- Kireeva, N., Pozdnyak, I., & Gazizulina, A. (2019b). Filling a Knowledge Base for Expert System in Information Security. *IOP Conference Series: Materials Science and Engineering*, 618(1), 12085. <https://doi.org/10.1088/1757-899X/618/1/012085>
- Klyne, G., & Carroll, J. (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax*.
- Knublauch, H. (2002). *An Agile Development Methodology for Knowledge-Based Systems Including a Java Framework for Knowledge Modeling and Appropriate Tool Support* (p. 216). Universität Ulm.
- Kolodner, J. (2014). *Case-based reasoning*. Morgan Kaufmann.
- Kong, H. K., Hong, M. K., & Kim, T. S. (2018). Security risk assessment framework for smart car using the attack tree analysis. *Journal of Ambient Intelligence and Humanized Computing*, 9(3), 531–551. <https://doi.org/10.1007/s12652-016-0442-8>
- Kontopoulos, E., Martinopoulos, G., Lazarou, D., & Bassiliades, N. (2016). An ontology-based decision support tool for optimizing domestic solar hot water system selection. *Journal of Cleaner Production*, 112, 4636–4646. <https://doi.org/10.1016/j.jclepro.2015.08.088>
- Kopena, J. B., & Regli, W. C. (2003). DAMLJessKB: A tool for reasoning with the Semantic Web. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2870, 628–643. https://doi.org/10.1007/978-3-540-39718-2_40
- Korać, D., & Simić, D. (2017). Design of fuzzy expert system for evaluation of contemporary user authentication methods intended for mobile devices. *Control Engineering and Applied Informatics*, 19(4), 93–100.
- Kordy, B., Kordy, P., Mauw, S., & Schweitzer, P. (2013). ADTool: Security analysis with attack-defense trees. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8054 LNCS(318003), 173–176. https://doi.org/10.1007/978-3-642-40196-1_15
- Kordy, B., Mauw, S., Radomirović, S., & Schweitzer, P. (2010). Foundations of attack–defense trees. *International Workshop on Formal Aspects in Security and Trust*, 80–95.
- Kordy, B., Piètre-Cambacédès, L., & Schweitzer, P. (2014). DAG-based attack and defense modeling: Don’t miss the forest for the attack trees. *Computer Science Review*, 13–14(C), 1–38. <https://doi.org/10.1016/j.cosrev.2014.07.001>
- Kornecki, A. J., & Liu, M. (2013). Fault tree analysis for safety/security verification in aviation software. *Electronics*, 2(1), 41–56. <https://doi.org/10.3390/electronics2010041>
- Kozhakhmet, K., Bortsova, G., Inoue, A., & Atymtayeva, L. (2012). Expert system for security audit using fuzzy logic. *CEUR Workshop Proceedings*, 841, 146–151.
- Kozubíková, L., Belás, J., Bilan, Y., & Bartoš, P. (2015). Personal characteristics of entrepreneurs in the context of perception and management of business risk in the SME segment. *Economics and Sociology*, 8(1), 41–54. <https://doi.org/10.14254/2071-789X.2015/8-1/4>

- Kruse, R., & Borgelt, C. (2003). Information mining. In *International Journal of Approximate Reasoning*, Vol. 32, Issues 2–3, pp. 63–65. IGI Global. [https://doi.org/10.1016/S0888-613X\(02\)00088-9](https://doi.org/10.1016/S0888-613X(02)00088-9)
- Kulvietis, G., Kulvietienė, R. ir Rudzkienė, V. (1996). Įvadas į dirbtinio intelekto ir ekspertinių sistemų kursą. Technika.
- Kumar, R., Schivo, S., Ruijters, E., Yildiz, B. M., Huistra, D., Brandt, J., Rensink, A., & Stoelinga, M. (2018). Effective analysis of attack trees: A model-driven approach. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10802 LNCS, 56–73. https://doi.org/10.1007/978-3-319-89363-1_4
- Kumar, S., & Prasad, R. (2015). Importance of Expert System Shell in Development of Expert System. *International Journal of Innovative Research & Development*, 4(3), 128–133. www.ijird.com
- Kunz, J. C., Stelzner, M. J., & Williams, M. D. (1989). From Classic Expert Systems to Models: Introduction to a Methodology for Building Model-Based Systems. In *Studies in Computer Science and Artificial Intelligence*, Vol. 5, Issue C, pp. 87–110. Elsevier, North-Holland. <https://doi.org/10.1016/B978-0-444-87321-7.50009-0>
- Langenhan, M. K., Leka, S., & Jain, A. (2013). Psychosocial risks: Is risk management strategic enough in business and policy making? *Safety and Health at Work*, 4(2), 87–94. <https://doi.org/10.1016/j.shaw.2013.04.003>
- Lashkarara, S. (2017). *Managing Security Risks Using Attack-Defense Trees*.
- Lethbridge, T. C., Sim, S. E., & Singer, J. (2005). Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10(3), 311–341. <https://doi.org/10.1007/s10664-005-1290-x>
- Li, D., Cai, Z., Deng, L., Yao, X., & Wang, H. H. (2019). Information security model of block chain based on intrusion sensing in the IoT environment. *Cluster Computing*, 22(1), 451–468. <https://doi.org/10.1007/s10586-018-2516-1>
- Liao, S. H. (2005). Expert system methodologies and applications-a decade review from 1995 to 2004. *Expert Systems with Applications*, 28(1), 93–103. <https://doi.org/10.1016/j.eswa.2004.08.003>
- Library of Attack Trees*. Security and Trust of Software Systems. <https://satoss.uni.lu/projects/atrees/library.php>
- Lindqvist, U., & Porras, P. A. (1999). Detecting computer and network misuse through the production-based expert system toolset (P-BEST). *Proceedings - IEEE Symposium on Security and Privacy*, 1999-Janua, 146–161. <https://doi.org/10.1109/SECPRI.1999.766911>
- Liu, D., Gu, T., & Xue, J. P. (2010). Rule engine based on improvement rete algorithm. *2010 International Conference on Apperceiving Computing and Intelligence Analysis, ICACIA 2010 - Proceeding*, 346–349. <https://doi.org/10.1109/ICACIA.2010.5709916>

- Löhner, B. (2018). Attack-Defense-Trees and other Security Modeling Tools. *Network Architectures and Services, Seminar Fu*(Seminars FI / IITM SS 18), 7.
- Lu, J., Bai, D., Zhang, N., Yu, T., & Zhang, X. (2016). Fuzzy case-based reasoning system. *Applied Sciences*, 6(7), 189.
- Malarvizhi, R., & Saraswathi, K. (2013). Web Content Mining Techniques Tools & Algorithms – A Comprehensive Study. *Web Content Mining Techniques Tools & Algorithms – A Comprehensive Study*, 4(8), 2940–2945.
- Maldoff, G. (2017). The Risk-Based Approach in the GDPR: Interpretation and Implications. *CIPP/US - IAPP Westin Fellow*, 12.
- Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1), 1–13. [https://doi.org/10.1016/S0020-7373\(75\)80002-2](https://doi.org/10.1016/S0020-7373(75)80002-2)
- Mankad, K. B. (2018). An expert system design to categorize multiple intelligence of students. *IUP Journal of Computer Sciences*, 9(3), 23–35.
- Maskeliunas, R., Damaševicius, R., & Segal, S. (2019). A review of internet of things technologies for ambient assisted living environments. *Future Internet*, 11(12), 259. <https://doi.org/10.3390/FI11120259>
- McLaughlin, S., Podkuiko, D., Miadzezhanka, S., Delozier, A., & McDaniel, P. (2010). Multi-vendor penetration testing in the advanced metering infrastructure. *Proceedings - Annual Computer Security Applications Conference, ACSAC*, 107–116. <https://doi.org/10.1145/1920261.1920277>
- Meditskos, G., & Bassiliades, N. (2008). Combining a DL reasoner and a rule engine for improving entailment-based OWL reasoning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5318 LNCS, 277–292. https://doi.org/10.1007/978-3-540-88564-1_18
- Miller, B. M. (1993). Object-oriented expert systems and their applications to sedimentary basin analysis. *US Geological Survey Bulletin*, 2048.
- Millet, R., & Fenwick, J. (2005). Decision support. In *IET Seminar Digest* (Vol. 2005, Issue 11083, pp. 205–219). Springer. <https://doi.org/10.1049/ic:20050434>
- Misiulienė, R. (2012). Daugkartinio ontologijų panaudojimo galimybės e. mokymesi. *Studijos šiuolaikinėje visuomenėje*, 3 (1). Šiaurės Lietuvos kolegija, 162–168.
- Mohamed, A. E. (2017). Comparative Study of Four Supervised Machine Learning Techniques for Classification. *International Journal of Applied Science and Technology*, 7(2), 5–18.
- Mohammad, S. M., & Lakshmisri, S. (2018). Security Automation in Information Technology. *International Journal of Creative Research Thoughts (IJCRT)*, Vol. 6.
- Moore, A. P., Ellison, R. J., & Linger, R. C. (2001). Attack Modeling for Information Survivability. *Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst*, 17(March), 15–33.

- Mujawar, I. K., & Jadhav, B. T. (2017). Comprehensive study on web based expert systems for disease diagnosis and treatment. *International Journal of Computer Engineering and Applications*, 11(October), 9.
- Mujawar, I. K., & Jadhav, B. T. (2019). Web-based Fuzzy Expert System for Diabetes Diagnosis. *International Journal of Computer Sciences and Engineering*, 7(2), 995–1000. <https://doi.org/10.26438/ijcse/v7i2.9951000>
- Nalepa, G. J., Bobek, S., Ligeza, A., & Kaczor, K. (2011). Algorithms for rule inference in modularized rule bases. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6826 LNCS, 305–312. https://doi.org/10.1007/978-3-642-22546-8_24
- Nalepa, G. J., & Kluza, K. (2012). UML representation for rule-based application models with XTT2-based business rules. *International Journal of Software Engineering and Knowledge Engineering*, 22(4), 485–524. <https://doi.org/10.1142/S021819401250012X>
- Negnevitsky, M. (2005). Artificial Intelligence: A Guide to Intelligent Systems. *Pearson Education Limited*, 2, 414.
- Nourian, R., Mousavi, S. M., & Raissi, S. (2019). A fuzzy expert system for mitigation of risks and effective control of gas pressure reduction stations with a real application. *Journal of Loss Prevention in the Process Industries*, 59, 77–90. <https://doi.org/10.1016/j.jlp.2019.03.003>
- Obrst, L., Chase, P., & Markeloff, R. (2014). Developing an ontology of the cyber security domain. *CEUR Workshop Proceedings*, 966, 49–56.
- Ogu, E. C., & Adekunle, Y. A. (2013). Basic Concepts of Expert System Shells and an Efficient Model for Knowledge Acquisition. *International Journal of Science and Research (IJSR)*, 2(4), 554–559.
- Olifer, D., Goranin, N., Cenys, A., Kaceniauskas, A., & Janulevicius, J. (2019). Defining the minimum security baseline in a multiple security standards environment by graph theory techniques. *Applied Sciences (Switzerland)*, 9(4), 681. <https://doi.org/10.3390/app9040681>
- Olifer, D., Goranin, N., Janulevicius, J., Kaceniauskas, A., & Cenys, A. (2018). Improvement of security costs evaluation process by using data automatically captured from BPMN and EPC models. *Lecture Notes in Business Information Processing*, 308, 698–709. https://doi.org/10.1007/978-3-319-74030-0_56
- Olson, J. R., & Rueter, H. H. (1987). Extracting expertise from experts: Methods for knowledge acquisition. *Expert Systems*, 4(3), 152–168. <https://doi.org/10.1111/j.1468-0394.1987.tb00139.x>
- Otero-Cerdeira, L., Rodríguez-Martínez, F. J., & Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2), 949–971.
- Owasp. (2013). *OWASP Risk Rating Methodology*. Owasp. https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

- Panchal, D., Singh, A. K., Chatterjee, P., Zavadskas, E. K., & Keshavarz-Ghorabae, M. (2019). A new fuzzy methodology-based structured framework for RAM and risk analysis. *Applied Soft Computing Journal*, 74, 242–254. <https://doi.org/10.1016/j.asoc.2018.10.033>
- Paulo, P. V., Guimaraes, A. J., Rezende, T. S., Araujo, V. S., Do Nascimento, L. A. F., & Batista, L. O. (2020). An Intelligent Hybrid Model for the Construction of Expert Systems in Malware Detection. *IEEE Conference on Evolving and Adaptive Intelligent Systems*, 2020-May, 1–8. <https://doi.org/10.1109/EAIS48028.2020.9122770>
- Peltier, T. R. (2004). Risk analysis and risk management. *Information Systems Security*, 13(4), 44–56. <https://doi.org/10.1201/1086/44640.13.4.20040901/83732.7>
- Peltier, T. R. (2005). Information security risk analysis, second edition. In *Information Security Risk Analysis, Second Edition*. CRC press. <https://doi.org/10.1201/9781420031195>
- Petrica, G., Axinte, S. D., Bacivarov, I. C., Firoiu, M., & Mihai, I. C. (2017). Studying cyber security threats to web platforms using attack tree diagrams. *Proceedings of the 9th International Conference on Electronics, Computers and Artificial Intelligence*, 1–6. <https://doi.org/10.1109/ECAI.2017.8166456>
- Pietre-Cambacedes, L., & Bouissou, M. (2010). Beyond attack trees: Dynamic security modeling with Boolean logic Driven Markov Processes (BDMP). *EDCC-8 - Proceedings of the 8th European Dependable Computing Conference*, 199–208. <https://doi.org/10.1109/EDCC.2010.32>
- Pinchinat, S., Acher, M., & Vojtisek, D. (2015). Towards synthesis of attack trees for supporting computer-aided risk analysis. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8938, 363–375. https://doi.org/10.1007/978-3-319-15201-1_24
- Powell, P. (1996). Information Management: The Evaluation of Information Systems Investments. In *The Journal of Strategic Information Systems*, Vol. 5, Issue 1. Springer. [https://doi.org/10.1016/s0963-8687\(96\)80025-4](https://doi.org/10.1016/s0963-8687(96)80025-4)
- Preston, S., Chapman, C., Pinfold, M., & Smith, G. (2005). Knowledge acquisition for knowledge-based engineering systems. *International Journal of Information Technology and Management*, 4(1), 1–11. <https://doi.org/10.1504/IJITM.2005.006401>
- Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences (Switzerland)*, 9(5), 909. <https://doi.org/10.3390/app9050909>
- Ralston, P. A. S., Graham, J. H., & Hieb, J. L. (2007). Cyber security risk assessment for SCADA and DCS networks. *ISA Transactions*, 46(4), 583–594. <https://doi.org/10.1016/j.isatra.2007.04.003>
- Ramanauskaite, S., Olifer, D., Goranin, N., & Čenys, A. (2013). Security ontology for adaptive mapping of security standards. *International Journal of Computers, Communications and Control*, 8(6), 878–890. <https://doi.org/10.15837/ijccc.2013.6.764>

- Reid, R., & Van Niekerk, J. (2014). From information security to cyber security cultures. *2014 Information Security for South Africa - Proceedings of the ISSA 2014 Conference*, 38, 97–102. <https://doi.org/10.1109/ISSA.2014.6950492>
- Rhem, A. J. (2005). *UML for developing knowledge management systems*. Auerbach Publications.
- Rick, U., Vossen, R., Richert, A., & Henning, K. (2010). Designing agile processes in information management. *ICIME 2010 - 2010 2nd IEEE International Conference on Information Management and Engineering*, 4, 156–160. <https://doi.org/10.1109/ICIME.2010.5477776>
- Rios-Boutin, J., Orriols-Puig, A., & Garrell-Guiu, J. M. (2008). Artificial data sets based on knowledge generators: Analysis of learning algorithms efficiency. *2008 Eighth International Conference on Hybrid Intelligent Systems*, 873–878.
- Rios, E., Rego, A., Iturbe, E., Higuero, M., & Larrucea, X. (2020). Continuous quantitative risk management in smart grids using attack defense trees. *Sensors (Switzerland)*, 20(16), 1–25. <https://doi.org/10.3390/s20164404>
- Ristoski, P., & Paulheim, H. (2016). Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Journal of Web Semantics*, 36, 1–22. <https://doi.org/10.1016/j.websem.2016.01.001>
- Ru, Y., Wang, Y., Li, J., Liu, J., Yang, G., Yuan, K., & Liu, K. (2016). Risk assessment of cyber attacks in ECPS based on attack tree and AHP. *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016*, 465–470. <https://doi.org/10.1109/FSKD.2016.7603218>
- Saini, V., Duan, Q., & Paruchuri, V. (2008). Threat modeling using attack trees. *Journal of Computing Sciences in Colleges*, 23(4), 124–131.
- Sajja, P. S., & Akerkar, R. (2010). Knowledge-Based Systems for Development. *Advanced Knowledge Based Systems: Model, Applications & Research*, 1, 1–11.
- Salter, C., Saydjari, O. S., Schneter, B., & Watlner, J. (1998). Toward a secure system engineering methodology. *Proceedings New Security Paradigms Workshop, Part F1292*, 2–10. <https://doi.org/10.1145/310889.310900>
- Schafer, B., & Keppens, J. (2004). Using the box to think outside it - creative scepticism and computer decision support in crime investigation. *Proceedings of the IVR 21st World Congress Special Workshop on Artificial Intelligence in the Law: Creativity in Legal Problem Solving*.
- Schmittling, R. A. M. (2010). Performing a Security Risk Assessment. *ISACA Journal*, 1, 1–7.
- Schneier, B. (2015). Attack Trees. *Secrets and Lies*, 24(12), 318–333. <https://doi.org/10.1002/9781119183631.ch21>
- Scott, A. M., Forbes, C., Clark, J., Carter, M., Glasziou, P., & Munn, Z. (2021). Systematic review automation tools improve efficiency but lack of knowledge impedes their adoption: a survey. *Journal of Clinical Epidemiology*, 138, 80–94.

- Shang, W., Gong, T., Chen, C., Hou, J., & Zeng, P. (2019). Information Security Risk Assessment Method for Ship Control System Based on Fuzzy Sets and Attack Trees. *Security and Communication Networks*, 2019. <https://doi.org/10.1155/2019/3574675>
- Shannon, R. (2012). *What is HTML?* The Rosen Publishing Group, Inc.
- Sheyner, O., & Wing, J. (2004). Tools for generating and analyzing attack graphs. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3188, 344–371. https://doi.org/10.1007/978-3-540-30101-1_17
- Shorabeh, S. N., Firozjaei, M. K., Nematollahi, O., Firozjaei, H. K., & Jelokhani-Niaraki, M. (2019). A risk-based multi-criteria spatial decision analysis for solar power plant site selection in different climates: A case study in Iran. *Renewable Energy*, 143, 958–973. <https://doi.org/10.1016/j.renene.2019.05.063>
- Shostack, A. (2014). *Threat Modeling : Designing for Security*. John Wiley & Sons.
- Sicilia, M. A., García-Barriocanal, E., Bermejo-Higuera, J., & Sánchez-Alonso, S. (2015). What are information security ontologies useful for? *Communications in Computer and Information Science*, 544, 51–61. https://doi.org/10.1007/978-3-319-24129-6_5
- Sikos, L. F. (2019). OWL ontologies in cybersecurity: Conceptual modeling of cyber-knowledge. In *Intelligent Systems Reference Library*, Vol. 151, pp. 1–17. Springer. https://doi.org/10.1007/978-3-319-98842-9_1
- Simiński, R., & Xięski, T. (2016). Physical knowledge base representation for web expert system shell. In *Communications in Computer and Information Science*, Vol. 613, pp. 558–570. Springer. https://doi.org/10.1007/978-3-319-34099-9_43
- Singh, B., & Kumar Singh, H. (2010). Web Data Mining research: A survey. *2010 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2010*, 661–670. <https://doi.org/10.1109/ICCIC.2010.5705856>
- Slimani, T. (2015). Ontology development: A comparing study on tools, languages and formalisms. *Indian Journal of Science and Technology*, 8(24), 1–12. <https://doi.org/10.17485/ijst/2015/v8i34/54249>
- Speck-Planche, A., & Kleandrova, V. V. (2020). Demystifying Artificial Neural Networks as Generators of New Chemical Knowledge: Antimalarial Drug Discovery as a Case Study. *Machine Learning in Chemistry*, 17, 398.
- Spinellis, D., Kokolakis, S., & Gritzalis, S. (1999). Security requirements, risks and recommendations for small enterprise and home-office environments. *Information Management and Computer Security*, 7(3–4), 121–128. <https://doi.org/10.1108/09685229910371071>
- Stowell, F. (1993). Introduction to Information Systems Stream. In *Systems Science*. John Wiley & Sons. https://doi.org/10.1007/978-1-4615-2862-3_55
- Stravinskienė, A., & Gudas, S. (2010). Duomenų gavyba paremta veiklos modeliui. *Information Technologies*, 2010, 16th.

- Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). *MITRE ATT&CK - Design and Philosophy*. The MITRE Corporation.
- Tankelevičienė, L., & Sakalauskas, R. (2008). Dalykinės srities ontologija grįsta intelektualioji gedimų diagnostikos sistema. *Jaunųjų mokslininkų darbai*, 5, 65–72.
- Tentilucci, M., Roberts, N., Kandari, S., Johnson, D., Bogaard, D., Stackpole, B., & Markowsky, G. (2015). Crowdsourcing Computer Security Attack Trees. *10th Annual Symposium on Information Assurance (ASIA '15)*, June 2-3, 2015, Albany, NY, 19–23.
- Tsudik, G., & Summers, R. (1990). AudES-An Expert System for Security Auditing. *Iaai*, 1–10.
- Turban, E., Aronson, J. E., & Liang, T. P.. (2018). *Decision Support Systems and Intelligent Systems ed 7* (Vol. 4). Pearson Prentice-Hall Upper Saddle River, NJ, USA:
- Venčkauskas, A., Damaševičius, R., Marcinkevičius, R., & Karpavičius, A. (2015). Problems of authorship identification of the national language electronic discourse. *Communications in Computer and Information Science*, 538, 415–432. https://doi.org/10.1007/978-3-319-24770-0_36
- Venkata Subba Reddy, P. (2017). Fuzzy logic based on Belief and Disbelief membership functions. *Fuzzy Information and Engineering*, 9(4), 405–422. <https://doi.org/10.1016/j.fiae.2017.12.001>
- Vigo, R., Nielson, F., & Nielson, H. R. (2014). Automated generation of attack trees. *Proceedings of the Computer Security Foundations Workshop, 2014-Janua*, 337–350. <https://doi.org/10.1109/CSF.2014.31>
- Vintr, Z., Valis, D., & Malach, J. (2012). Attack tree-based evaluation of physical protection systems vulnerability. *Proceedings - International Carnahan Conference on Security Technology*, 59–65. <https://doi.org/10.1109/CCST.2012.6393538>
- Wagner, C. (2006). Breaking the knowledge acquisition bottleneck through conversational knowledge management. *Information Resources Management Journal*, 19(1), 70–83. <https://doi.org/10.4018/irmj.2006010104>
- Wang, F., Reinhardt, W., & Reinhardt, W. (2005). Using ontologies for integrity constraint definition. *Proceedings of the 4th International Symposium On Spatial Data Quality*, 5, 304–313.
- Wang, P., Lin, H. T., Wang, T. C., Lin, W. H., & Kuo, P. T. (2012). Threat risk analysis for cloud security based on Attack-Defense Trees. *International Journal of Advancements in Computing Technology*, 4(17), 607–617. <https://doi.org/10.4156/ijact.vol4.issue17.70>
- Waszkowski, R., Nowicki, T., & Worwa, K. (2018). Corporate efficiency improvement with business process automation. *MATEC Web of Conferences*, 210, 2012.
- Wei, B., Sugiura, N., & Maekawa, T. (2001). Use of artificial neural network in the field of security. *Water Res.*, 35(8), 2022–2028.
- Wu, X., Chen, H., Wu, G., Liu, J., Zheng, Q., He, X., Zhou, A., Zhao, Z. Q., Wei, B., Gao, M., Li, Y., Zhang, Q., Zhang, S., Lu, R., & Zheng, N. (2015). Knowledge Engineering

- with Big Data. *IEEE Intelligent Systems*, 30(5), 46–55. <https://doi.org/10.1109/MIS.2015.56>
- Wygant, R. M. (1989). CLIPS - A powerful development and delivery expert system tool. *Computers and Industrial Engineering*, 17(1–4), 546–549. [https://doi.org/10.1016/0360-8352\(89\)90121-6](https://doi.org/10.1016/0360-8352(89)90121-6)
- Wynyard. (2015). *Wynyard Risk Management*. <http://docplayer.net/2971103-Wynyard-risk-management-powerful-software-connecting-the-dots.html>
- Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *ArXiv Preprint ArXiv:1708.07747*.
- Xiao, Y., Wang, Y. J., & Huang, Z. G. (2012). Survivability analysis of SOA based on attack tree models. *International Conference on Communication Technology Proceedings, ICCT*, 819–823. <https://doi.org/10.1109/ICCT.2012.6511317>
- Xięski, T., & Simiński, R. (2017). A performance study of two inference algorithms for a distributed expert system shell. *Communications in Computer and Information Science*, 716, 512–526. https://doi.org/10.1007/978-3-319-58274-0_40
- Yazdani, M., Alidoosti, A., & Basiri, M. H. (2011). Risk Analysis for Critical Infrastructures Using Fuzzy TOPSIS. *Journal of Management Research*, 4(1), 27–40. <https://doi.org/10.5296/jmr.v4i1.979>
- Young, C. (2020). *A Critical Study into the Effective Application and Use of Information Security Risk Management Frameworks by Large Organisations in the UK A Critical Study Into the Effective Application and Use of Information Security Risk Management Frameworks by Large*.
- Yu, L. (2011). A developer's guide to the semantic web. In *A Developer's Guide to the Semantic Web*. Springer Science & Business Media. <https://doi.org/10.1007/978-3-642-15970-1>
- Yurin, A. Y., & Dorodnykh, N. O. (2020). Personal knowledge base designer: Software for expert systems prototyping. *SoftwareX*, 11, 100411. <https://doi.org/10.1016/j.softx.2020.100411>
- Zavadskas, E. K., Turskis, Z., Vilutienė, T., & Lepkova, N. (2017). Integrated group fuzzy multi-criteria model: Case of facilities management strategy selection. *Expert Systems with Applications*, 82, 317–331. <https://doi.org/10.1016/j.eswa.2017.03.072>

List of Scientific Publications by the Author on the Topic of the Dissertation

Papers in the Reviewed Scientific Journals

Vitkus, D., Jezukeviciute, J., & Goranin, N. (2020). Dynamic Expert System-Based Geographically Adapted Malware Risk Evaluation Method. *International Journal of Computers, Communications & Control*, 15(3). <https://doi.org/10.15837/ijccc.2020.3.3865>

Vitkus, D., Salter, J., Goranin, N., & Čeponis, D. (2020). Method for Attack Tree Data Transformation and Import into IT Risk Analysis Expert Systems. *Applied Sciences*, 10(23), 8423. <https://doi.org/10.3390/app10238423>

Vitkus, D., Steckevicius, Z., Goranin, N., Kalibatiene, D., & Cenys, A. (2019). Automated expert system knowledge base development method for information security risk analysis. *International Journal of Computers, Communications and Control*, 14(6), 743–758. <https://doi.org/10.15837/ijccc.2019.6.3668>

Summary in Lithuanian

Įvadas

Problemos formulavimas

Informacijos saugos rizikų vertinimas gali padėti organizacijoms valdyti informacijos saugą (Agrawal, 2017). Tai padeda identifikuoti ne tik silpnąsias informacijos saugos vietas organizacijoje, bet taip pat gali padėti mažinti informacijos saugos užtikrinimo kaštus, kadangi nėra poreikio diegti visas įmanomas informacijos saugos kontrolės priemones, bet tik tas, kurios identifikuotos kaip būtinos, siekiant suvaldyti kylančias rizikas.

Organizacijos turi du būdus, kaip atlikti informacijos saugos rizikų analizę: samdyti specialistus (ar įmones) iš išorės arba įdarbinti specialistus. Abu pasirinkimai yra gana brangūs. Keletas autorių (Janulevičius & Čenys, 2014; Kailay & Jarratt, 1995; Kozhakhmet et al., 2012) informacijos saugos rizikos analizei siūlo naudoti ekspertines sistemas. Ekspertinėms sistemoms pagal savo struktūrą reikalinga žinių bazė, kuri būtų laiku atnaujinta ir būtų suformuota kokybiškais duomenimis.

Ekspertinių sistemų žinių bazės suformavimas ir atnaujinimas tampa problema, kadangi tai yra brangus ir ilgai trunkantis procesas. Tai rodo, kad ekspertinių sistemų žinių bazės formavimo ir atnaujinimo automatizavimas yra svarbus. Kita vertus, egzistuoja laisvai prieinamų susistemintų žinių šaltinių, kurie gali būti panaudojami automatizuojant informacijos saugos rizikų analizės ekspertinių sistemų žinių bazes. Automatizuotas tokių žinių konvertavimas į ekspertinių sistemų žinių bazes leistų efektyviai ir mažesniais kaštais atlikti informacijos saugos rizikos vertinimą.

Darbo aktualumas

Įvairūs autoriai (Lashkarara, 2017; Ralston et al., 2007; Shang et al., 2019) sutinka, kad šiomis dienomis informacija yra viena iš kritinių organizacijos dalių, nepriklausomai nuo jos dydžio. Net pati mažiausia organizacija turi informacijos, kurią reikia saugoti (Stowell, 1993). Informacijos saugos rizikų analizė gali padėti identifikuoti silpnąsias vietas, kurias reikia saugoti.

Didelės organizacijos gali sau leisti samdytis informacijos saugos rizikų valdymo specialistus, tačiau mažos ir vidutinio dydžio įmonės tai leisti gali ne visada. Ekspertinės sistemos gali padėti atlikti informacijos saugos rizikų analizę, tačiau ekspertinių sistemų žinių bazės atnaujinimas tampa problema. Automatizavus informacijos saugos rizikų analizės ekspertinių sistemų žinių bazių formavimą ir atnaujinimą, būtų galima automatizuoti informacijos saugos rizikų analizės procesą, ypač mažose ir vidutinio dydžio įmonėse.

Žinių bazių automatizavimo tema yra palyginti nauja, nors keletas autorių jau yra pasiūlę automatizavimo metodų, kurie pademonstravo ne tik automatizavimo galimybes, bet ir naudas. Pasiūlius metodus, kurie supaprastina rizikos analizės procesą naudojant ekspertines sistemas, rizikos analizė gali būti prieinamesnė mažoms ir vidutinėms įmonėms, kartu būtų padidintas rizikos vertinimo proceso tikslumas (Gadyatskaya, Harpes, et al., 2016; Janulevičius & Čenys, 2014) visiems, ypač mažoms ir vidutinėms įmonėms, kurios negali sau leisti samdyti aukštos kokybės saugumo ekspertų rizikos analizei.

Tyrimo objektas

Pagrindinis šio tyrimo objektas yra ekspertinių sistemų, naudojamų informacijos saugos rizikos analizei, žinių bazės pildymo automatizavimo metodai.

Darbo tikslas

Disertacijos tikslas – patobulinti informacijos saugos rizikos analizės ekspertinių sistemų žinių bazės generavimo būdus, kurie leistų sumažinti informacijos saugos valdymo atskirtį smulkaus ir vidutinio dydžio įmonėse, panaudojant esamus informacijos šaltinius.

Darbo uždaviniai

1. Atlikti esamų ekspertinių sistemų, skirtų informacijos saugos rizikos analizei, ir jų žinių bazės formavimo metodų analizę, nustatyti galimus informacijos šaltinius ir pateikti jų taikymo sritis.
2. Pasiūlyti metodus, kaip patobulinti automatizuotus informacijos saugos rizikos analizės ekspertinių sistemų žinių bazės pildymo metodus, naudojant esamus informacijos saugumo šaltinius.
3. Atlikti eksperimentinius bandymus ir išbandyti siūlomus metodus, skirtus automatizuotam ekspertinių sistemų žinių bazės pildymui, įvertinti gautus rezultatus ir apibrėžti būsimo darbo tikslus.

Tyrimų metodika

Tiriant darbo objektą, buvo pasirinkti toliau aprašyti tyrimo metodai:

- Teorinis (analizės ir sintezės) tyrimas atliktas siekiant nustatyti problemos sprendimo strategiją.
- Klasifikavimas: literatūroje pateiktų metodikų, privalumų ir trūkumų apibendrinimas, siekiant išgryninti disertacijos tyrimo objekto bei darbo tikslą.
- Patirtis: problemos sprendimas grindžiamas moksliniais tyrimais bei kitų mokslininkų įžvalgomis bei patirtimi.
- Eksperimentas: suformuota hipotezė patvirtinta eksperimentiniu tyrimu.
- Įvertinimas: išvados pateiktos atsižvelgiant į tyrimo metu surinktus išanalizuotus ir išaiškintus duomenis.

Darbo mokslinis naujumas

Darbo mokslinis naujumas pagrįstas šiais rezultatais:

1. Pasiūlytas naujas metodas automatizuotai transformuoti esamas informacijos saugos ontologijas į ekspertinių sistemos žinių bazės taisykles, metodas taikomas informacijos saugos rizikos analizei atlikti. Skirtingai nuo kitų metodų, jis tiesiogiai neintegruoja ontologijos, bet naudoja ontologijos elementų transformaciją į papildomas sistemos taisykles.
2. Pasiūlytas naujas metodas automatizuotai ekspertinių sistemų žinių bazei generuoti, jis paremtas regioninio kenkėjiškų programų paplitimo duomenų integravimu iš žiniatinklio realiojo laiko žemėlapiu, kuriame pateikiama naujausia informacija apie naujai atrastas grėsmes. Naudojant realiojo laiko žemėlapiu informaciją galima tiksliau apskaičiuoti informacijos saugos riziką konkrečiame regione.
3. Pasiūlytas naujas metodas, paremtas atakos medžių transformavimu į ekspertinių sistemų žinių bazės taisykles, turinčias informaciją apie atakos tikimybę ir kainą, siekiant palengvinti informacijos saugos rizikos analizę.

Darbo rezultatų praktinė reikšmė

Disertacijos pasiūlytų metodų teorinė ir praktinė svarba pasižymi jų pritaikomumu nuolat besikeičiančiame informacijos saugos valdymo sprendimų priėmimo procese, pagrįstu informacijos saugos rizikų valdymu, kuris apima ir privalomų reikalavimų įgyvendinimą.

Siūlomi informacijos saugos rizikos analizės ekspertinių sistemų žinių bazės formavimo metodai, naudojant esamas informacijos saugos ontologijas, kenkėjiškų programų paplitimo ir atakų medžių informaciją, demonstruoja tarpdalykinį požiūrį, derinantį tiek informatikos inžinerijos, tiek valdymo metodus, skirtus rizika pagrįstai informacijai spręsti.

Sukurtus įrankius, skirtus automatizuoti ontologijų, atakų medžių ir kenkėjiškų programų paplitimo informacijos transformavimui ir integravimui į informacijos saugos rizikų analizės ekspertinių sistemų žinių bazes, įmonės, ypač mažos ir vidutinės, gali tiesiogiai naudoti, kad sumažintų laiko ir žmoniškųjų išteklių sąnaudas, bei informacijos saugos rizikos analizę atlikti neturėdamos specialių informacijos saugos žinių.

Sukurtas ekspertinės sistemos prototipas parodė praktinį automatizuoto žinių bazės panaudojimo pritaikomumą informacijos saugos rizikos analizės procese.

Ginamieji teiginiai

1. Įvairių informacijos šaltinių, pasižyminčių skirtingomis savybėmis, transformavimas į ekspertinių sistemų taisykles ir jų įtraukimas į žinių bazę itin padidina aktualios informacijos kiekį, kuriuo operuoja ekspertinė sistema.
2. Automatiniu būdu suformuotos ekspertinių sistemų taisyklės, integruotos į žinių bazę, papildo ją informacija, reikalinga informacijos saugos rizikos analizei atlikti, ir yra darnios su rankiniu būdu suformuotomis taisyklėmis.
3. Dėl informacijos saugos rizikos analizės kompleksiskumo šiuo metu nėra vieno šaltinio, kuris būtų pakankamas ekspertinės sistemos žinių bazei formuoti. Skirtingi informacijos šaltiniai ekspertinės sistemos žinių bazę papildo specifine informacija, reikalinga informacijos saugos rizikai valdyti.

Darbo rezultatų aprobavimas

Disertacijos tema paskelbtos 3 žurnaluose, įtrauktuose į *Clarivate Analytics* (buv. *Thomson Reuters*) *Web of Science* duomenų bazę ir turinčiuose citavimo rodiklį. Moksliniai rezultatai buvo pristatyti 4 mokslinėse konferencijose, iš kurių 2 – tarptautinės:

- *The 24th International Nordic Conference on Secure IT Systems (NordSec 2019)*, Aalborg, Denmark.
- *The 11th International Workshop Data Analysis Methods for Software Systems (DAMSS 2019)*, Druskininkai, Lithuania.
- *The 22nd Junior Researchers Conference “Science – Future of Lithuania” on IT Security and Information Systems*, 2019, Vilnius, Lithuania.
- *The 23rd Junior Researchers Conference “Science – Future of Lithuania” on IT Security and Information Systems*, 2020, Vilnius, Lithuania.

Disertacijos struktūra

Darbą sudaro bendroji charakteristika, penki pagrindiniai skyriai, išvados, literatūros sąrašas, autoriaus publikacijų disertacijos tema sąrašas ir priedai. Disertacijos apimtis (be priedų) – 146 puslapiai, 54 iliustracijos ir 12 lentelių.

1. Ekspertinių sistemų, skirtų informacijos saugos rizikos analizei, ir jų žinių bazės pildymo metodų analizė

Šiame skyriuje apžvelgiama informacijos saugos rizikų analizės ekspertinių sistemų ir jų žinių bazių generavimo automatizavimo metodų analizė. Analizė pradedama informacijos saugos rizikų valdymo proceso paaiškinimu, kuriame išdėstomi pagrindiniai šio proceso elementai. Vėliau pateikiama pagrindinė ekspertinės sistemos architektūra ir jos aprašymas. Pateikiama esamų ekspertinių sistemų, skirtų informacijos saugos rizikų analizei, apžvalga. Analizuojant esamus žinių bazės generavimo būdus, pateikiami skirtingi žinių bazės generavimo būdai. Siekiant įvertinti automatizuoto žinių bazės generavimo galimybes, analizuojami galimi informacijos šaltiniai, kurie gali būti tinkami.

Informacija kiekvienoje šiandienos organizacijoje atlieka svarbų vaidmenį. Net mažiausia organizacija turi informacijos, kurią reikia saugoti. Informacijos saugos rizikų valdymas gali padėti organizacijoms sumažinti kylančias informacijos saugos rizikas iki priimtino lygio.

Informacijos saugos rizikų valdymas – procesų ir susijusių įrankių rinkinys, padedantis organizacijoms proaktyviai nustatyti grėsmes, silpnąsias vietas, įvertinti potencialų poveikį ir jo tikimybę (Young, 2020). Rizikai įvertinti reikalingos dvi pagrindinės dedamosios: galimas poveikis ir rizikos tikimybė. Svarbu pažymėti, kad rizikų valdymo procesas yra tęstinis. Kiekvienas rizikos suvaldymas reikalauja rizikos pervertinimo, todėl rizikų valdymas yra atliekamas periodiškai.

Ekspertinės sistemos yra viena iš pirmųjų dirbtinio intelekto formų (Negnevitsky, 2005). Vienas svarbiausių ekspertinės sistemos elementų – žinių bazė. Nuo jos kokybės priklauso ekspertinės sistemos rezultatai. Analizuojant mokslinę literatūrą nustatyti įvairūs bandymai, kaip palengvinti žinių bazių pildymą, tačiau nei vienas iš siūlytų nėra tinkamas informacijos saugos rizikų analizės ekspertinėms sistemoms.

Informacijos saugos rizikų analizė – sritis, reikalaujanti plačios ir tikslios analizės. Atlikus potencialių informacijos šaltinių, kurie galėtų būti naudojami automatizuotai formuoti ekspertinės sistemos žinių bazę, analizę nustatyta, kad vienas šaltinis nebūtų pakankamas, todėl buvo atrinkti trys šaltiniai, turintys skirtingas charakteristikas ir žinių bazę galintys papildyti įvairia, informacijos saugos rizikų analizei reikalinga informacija. Iš 8 potencialių šaltinių pagal atrinktus kriterijus buvo pasirinkti trys: informacijos saugos ontologijos, atakų medžiai, žiniatinklio duomenys (juos sudaro informacija apie kenkėjiško kodo paplitimą regionuose).

Informacijos saugos ontologijos yra vienas iš tinkamų informacijos šaltinių ekspertinių sistemų žinių bazės formavimui automatizuoti. Egzistuoja pakankamai laisvai prieinamų informacijos saugos ontologijų (Gao et al., 2013). Ontologijos išskiriamos kaip pakankamai tikslūs informacijos šaltiniai, turintys mažai klaidų. Iki šiol pasiūlyti ontologijų panaudojimo ekspertinėse sistemose metodai paremti jų integravimu į ekspertines sistemas, o ontologijų transformavimas į žinių bazę pasiūlytas tik specifinėms ontologijoms ir tik į specifines ekspertines sistemas (Otero-Cerdeira et al., 2015).

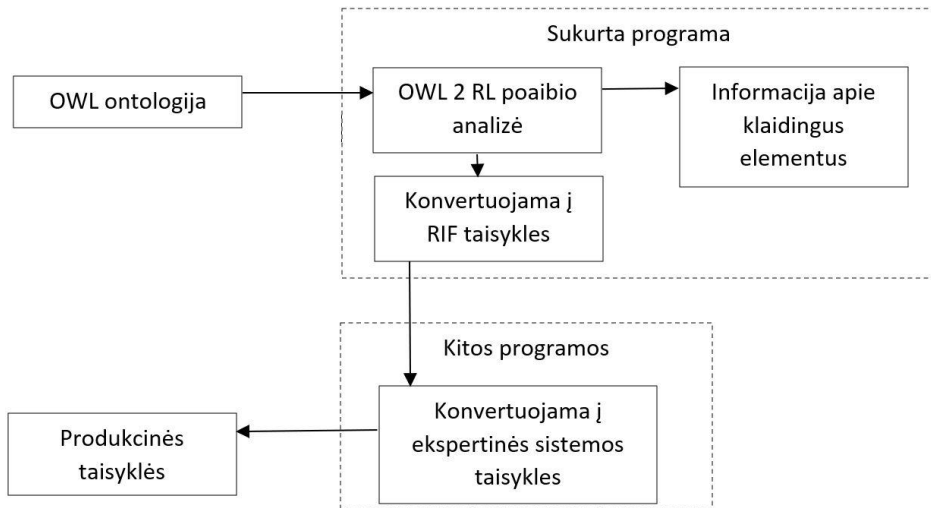
Žiniatinklio duomenys, kaip ekspertinės sistemos žinių bazės šaltinis, ypač informacijos saugos rizikų analizės srityje, yra tinkami, nes tai šaltinis, kuriame pirmiausiai pateikiama informacija apie kenkėjišką kodą (Singh & Singh, 2010). Žiniatinklyje laisvai prieinami įvairūs kenkėjiško kodo aprašai, pažeidžiamumai ir atakų aprašymo šaltiniai. Šiame darbe pasirinkta realiuoju laiku generuojama informacija apie kenkėjišką kodą ir jo paplitimą pagal regionus.

Trečiasis pasirinktas žinių šaltinis – atakos medžiai. Atakos medžiai skirti darbui su įvairaus tipo atakomis, jų analize. Kiekviename atakos medyje aprašomi konkrečios atakos vektoriai. Atakos medžiai naudingi informacijos saugos rizikos analizei, nes juose gali būti saugoma įvairi aktuali informacija, pvz., kontrolės priemonės, atakos sudėtingumas, kaina, tikimybė (Fila & Wideł, 2020). Taip pat yra pakankamai prieinamų atakos medžių tekstiniu formatu (Shang et al., 2019), o tai labai palengvina transformacijos uždavinį.

2. Automatizuotas ontologijų transformavimo į ekspertinės sistemos žinių bazę metodas

Šiame skyriuje apžvelgiamas automatizuoto saugos ontologijų transformavimo į informacijos saugos ekspertinių taisyklių žinių bazės taisyklės metodo aprašymas ir įgyvendinimo rezultatai. Skirtingai nei kiti, iki šiol pasiūlyti metodai, jis neintegruoja ontologijų tiesiogiai į ekspertines sistemas, tačiau konvertuoja į joms suprantamas taisykles. Tai leidžia į žinių bazę transformuoti kelias ontologijas, taip pat atnaujinti žinių bazę. Pasiūlytas metodas ir jo rezultatai buvo publikuoti tarptautiniame žurnale (Vitkus et al., 2019).

Ontologijų transformavimas į ekspertinių sistemų žinių bazės taisyklės įgyvendintas transformuojant *OWL 2 RL* poaibio ontologijas į *RIF* taisykles. Gautos *RIF* taisyklės, panaudojant jau egzistuojančius įrankius, buvo konvertuotos į *JESS* pagrindu sukurtos ekspertinės sistemos žinių bazės taisyklės (S2.1 pav.).



S2.1 pav. Bendrasis pasiūlyto metodo vaizdas

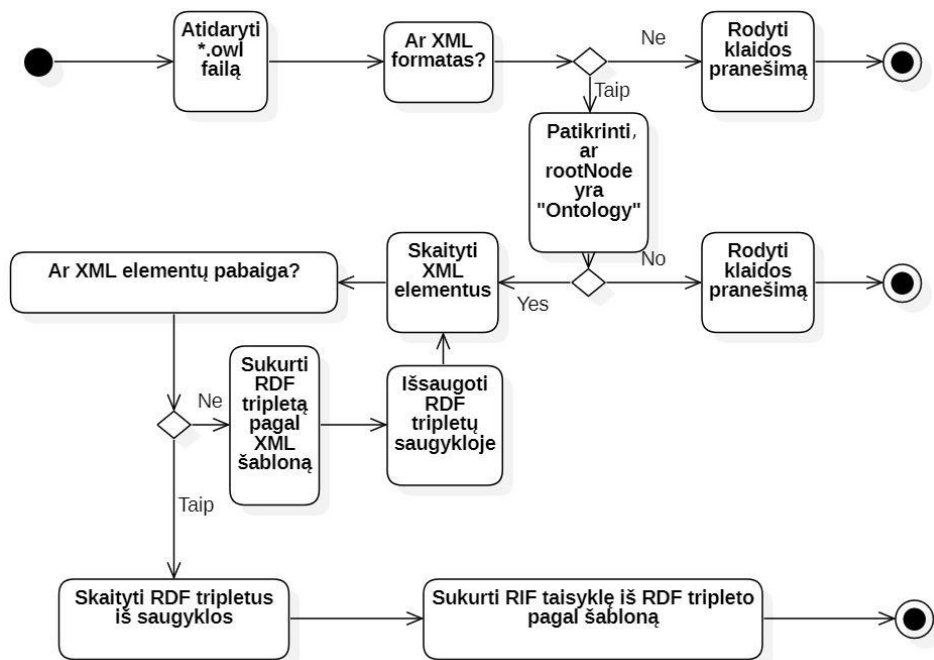
Pagrindiniai metodo taikymo etapai yra šie:

1. Pirmame etape *OWL 2 RL* poaibio ontologija transformuojama į *RIF* taisykles, išskiriant ir netransformuotinus elementus.
2. Antrajame etape gautos *RIF* taisyklės konvertuojamos į *JESS* ekspertinės sistemos žinių bazės taisykles.

Gautos *RIF* taisyklės gali būti konvertuojamos ir į kitų ekspertinių sistemų žinių bazės taisykles, pvz., *XSB*, *ASP*, arba naudojamos tiesiogiai, jeigu ekspertinė sistema palaiko *RIF* formatą.

OWL 2 RL transformacija į *RIF* paremta ontologijų išskaidymu į *RDF* tripletus, kurie, pritaikant konvertavimo šablonus, transformuojami į *RIF* taisykles (S2.2 pav.). *OWL 2 RL* transformavimo į *RIF* taisykles algoritmo pagrindiniai etapai yra šie:

1. Formatų tinkamumo patikrinimas. Ontologijos nuskaitymo pradžioje yra patikrinama, ar pateiktas failas yra *OWL/XML* formato. Turint *OWL* ontologiją, užrašytą kitokiu formatu, ją iš pradžių reikia transformuoti į *OWL/XML* formatą. Tam jau yra sukurti ir prieinami įvairūs įrankiai.
2. *RDF* tripletų sudarymas. Skaitymo metu elementai iš karto yra analizuojami. Atpažinus *OWL 2 RL* elementą, jis yra išsaugomas *RDF* tripletų formatu (subjektas, predikatas, objektas), kadangi iš tokio formato yra lengviausiai sukurti taisyklės. Neišskiriamai susijusius elementus gali sudaryti daugiau nei vienas tripletas. Tokie elementai saugomi sudėtinių *RDF* tripletų saugykloje.
3. *RDF* tripletai transformuojami į *RIF* taisyklės. Tai daroma iš anksto paruoštais šablonais, kurie užpildomi reikšmėmis, gaunamomis analizuojant ontologiją. Gautos taisyklės visiškai atitinka *RIF Core* formatą.



S2.2 pav. OWL 2 RL transformavimo į RIF algoritmas

Atliekant metodo eksperimentinius bandymus iš viso buvo transformuota 17 ontologijų į *RIF* taisyklės, t. y. 15 pavyzdinių W3C konsorciumo ontologijų bei 2 praktinės ontologijos: debesų kompiuterijos saugos ontologija ir informacijos saugos standartų ontologija (S2.3 lentelė).

S2.3 lentelė. Ontologijų transformavimo į ES žinių bazės taisyklės rezultatai

Ontologija	Sėkmingai transformuotų taisyklių dalis
W3C testavimo ontologijos (iš viso 15), atitinkančios OWL 2 RL	90 % (vidurkis)
Debesų kompiuterijos saugos ontologija	64 %
Informacijos saugos standartų ontologija	60 %

Suformuotų taisyklių dalis labai priklauso nuo pačios ontologijos. Atliekant testus su W3C ontologijomis, buvo parinktos ontologijos, atitinkančios OWL 2 RL profilį, dėl to iš jų pavyko gauti daugumą taisyklių, tačiau, paėmus praktinę ontologiją, dalis elementų neatitiko OWL 2 RL poaibio, todėl dalis elementų nebuvo atpažinti.

Metodo eksperimentinio bandymo metu W3C ontologijos transformuotos tik testavimo tikslais, o kitos dvi ontologijos pritaikytos praktiškai. Atlikus transformacijas, automatiškai buvo sukurta 601 RIF taisyklė iš Informacijos saugos standartų ontologijos ir 61 taisyklė iš Debesų kompiuterijos saugos ontologijos, iš viso 662 automatiškai transformuotos taisyklės iš abiejų ontologijų. Šios taisyklės buvo importuotos į informacijos saugos rizikos analizės eksperimentinės sistemos prototipą, pristatomą 5 skyriuje.

3. Kenkėjiško kodo pasiskirstymo pagal regionus duomenų integravimas į ekspertinės sistemos žinių bazę geografiškai priklausančioms rizikoms vertinti

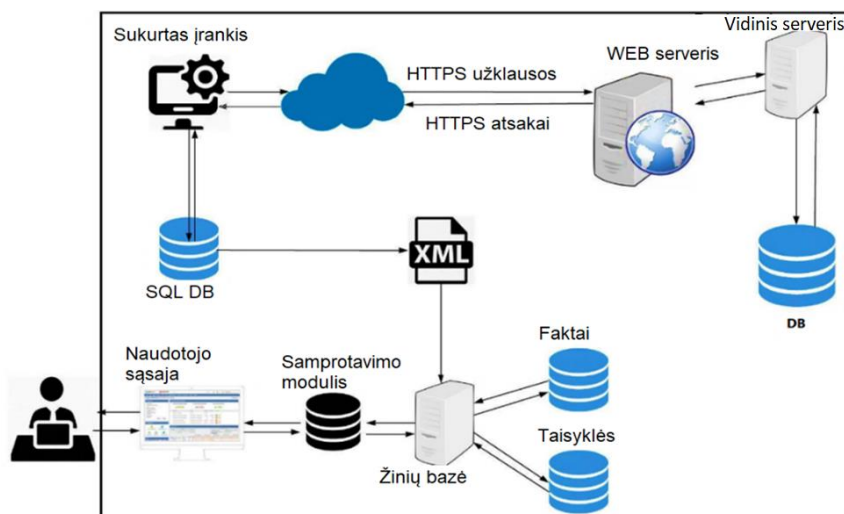
Šiame skyriuje apžvelgiamas pasiūlytas ir įgyvendintas žiniatinklio duomenų transformavimo į informacijos saugos analizės ekspertinių sistemų žinių bazės taisyklės metodas. Pasiūlytas metodas generuoja žinių bazės taisykles pagal kenkėjiško kodo aprašus ir jų pasiskirstymą regionuose, aprašai pateikti realiojo laiko atakų žemėlapiuose. Pasiūlytas metodas ir jo rezultatai publikuoti tarptautiniame žurnale (Vitkus, Jezukeviciute et al., 2020).

Metodas yra paremtas informacijos surinkimu iš kibernetinių atakų realiojo laiko žemėlapiu, kuriame pateikiama informacija apie kenkėjiškas programas, jų paplitimą regionuose. Metodui įgyvendinti pasirinktas *Kaspersky Lab Cyber-Threat Real-Time Map*. Ekspertinės sistemos žinių bazės taisyklės yra formuojamos išgaunant informaciją iš žiniatinklio, nuskaitant informaciją iš HTML žymų (S3.1 pav.).

Pagrindiniai pasiūlyto metodo taikymo etapai yra šie:

1. Siunčiamos HTTPS užklausos į žiniatinklį.
2. Gaunamas atsakymas iš žiniatinklio.
3. Analizuojamos HTML žymos, iš kurių reikiama informacija saugoma SQL duomenų bazėje.
4. SQL duomenų bazėje esanti informacija konvertuojama į JESS ekspertinei sistemai suprantamą XML formatą.

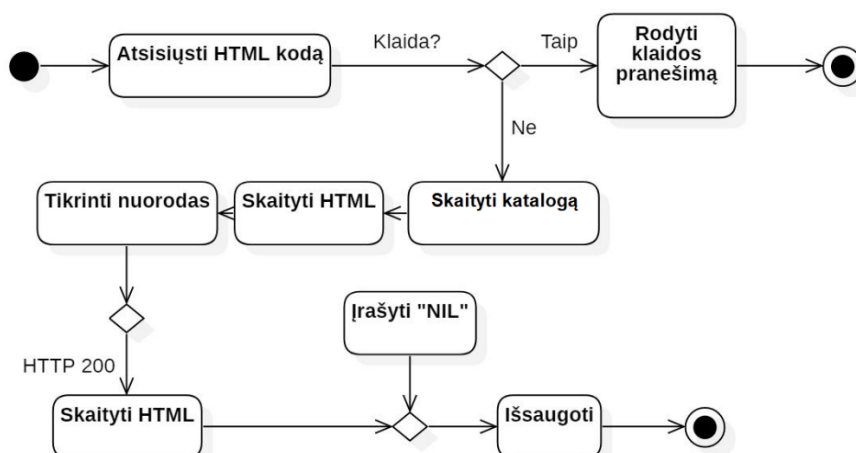
Sukurtas XML failas importuojamas į JESS pagrindu sukurtą ekspertinę sistemą.



S3.1 pav. Metodo architektūra

Metodas skaito žiniatinklio *HTML* kodą, ieško aprašytų žymų, o, juos aptikęs, juose esančią informaciją išsaugo (S3.2 pav.).

Atliekant metodo eksperimentinius bandymus iš viso buvo nuskaitytos 225 žiniatinklio žymos, iš kurių suformuotas 201 ekspertinės sistemos žinių bazės įrašas. 24 žymos nebuvo nuskaitytos dėl žiniatinklio klaidų: žymos buvo tuščios arba nuorodos vedė į neegzistuojančius puslapius. Eksperimentas rodo, kad iš principo automatinis duomenų išgavimas iš žiniatinklio svetainių ir jų transformavimas į ekspertinių sistemų taisykles yra įmanomas.



S3.2 pav. Transformacijos iš HTML algoritmas

Automatiškai gautos taisyklės buvo importuotos į *JESS* pagrindu sukurtą informacijos saugos rizikų analizės ekspertinę sistemą, kurioje atliktas informacijos saugos rizikos vertinimas naudojant šias žinių bazės taisykles. Rizikos įvertiniai apskaičiuoti pagal formulę $Rizika = Tikimybė * Poveikis$. Eksperimento metu buvo renkama informacija apie kenkėjiško kodo paplitimą trijose valstybėse: Lietuvoje, Latvijoje, Estijoje (S3.3 pav.).

```
*****
Impact level: 8.0
*****
```

Risk level (LT):	160
Risk level (LV):	97
Risk level (EST):	104
Platform:	Multi
Percentage of spread (LT):	20.1
Percentage of spread (LV):	12.1
Percentage of spread (EST):	13.04
Virus Name:	DangerousObject.Multi.Generic
Type:	On-Access Scan
Class:	DangerousObject
Description:	Malicious software is detected by KL Cloud

S3.3 pav. Kenkėjiško kodo rizikos vertinimo pavyzdys Lietuvoje, Latvijoje, Estijoje

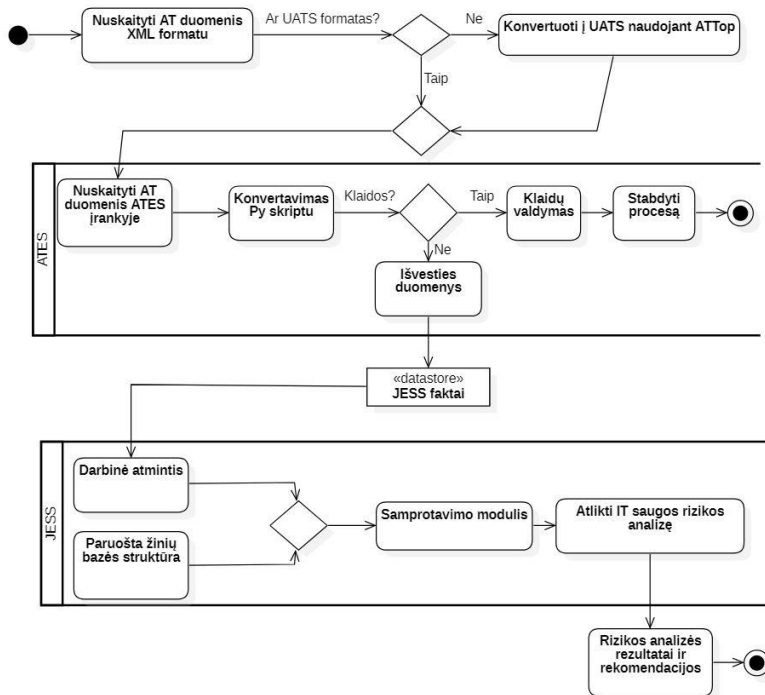
Pateiktas pavyzdys rodo, kad kenkėjiškas kodas gali būti skirtingai paplitęs skirtingose valstybėse, todėl ir rizikos lygis yra skirtingas, esant tam pačiam poveikiui.

Pagrindinis pasiūlyto ir įgyvendinto metodo privalumas yra galimybė turėti nuolat atnaujinamą ekspertinės sistemos žinių bazę, kurioje pateikiama informacija apie dabartines grėsmes, veikimo platformą, paplitimą regionuose. Kadangi metodas paremtas *HTML* žymų skaitymu, todėl jų pasikeitimas paveiktų metodo veikimą. Pasikeitus žiniatinklyje naudojamų *HTML* žymų pavadinimams reiktų atnaujinti jų aprašus ir metodą įgyvendinančiame įrankyje.

4. Atakos medžių duomenų transformavimo ir importavimo į informacijos saugos rizikų analizės ekspertinę sistemą metodas

Šiame skyriuje apžvelgiamas pasiūlytas ir įgyvendintas atakos medžių transformavimo į informacijos saugos rizikų analizės ekspertinės sistemos žinių bazę metodas. Pasiūlytas metodas transformuoja *UATS XML* užrašytus atakos medžius į *JESS* pagrindu sukurtos ekspertinės sistemos žinių bazės taisykles. Pasiūlytas metodas ir jo rezultatai publikuojami tarptautiniame žurnale (Vitkus, Salter et al., 2020).

Pasiūlytam metodui įgyvendinti buvo sukurtas *ATES* įrankis, kuris konvertuoja atakos medžius į *JESS* ekspertinės sistemos žinių bazės taisykles (S4.1 pav.).



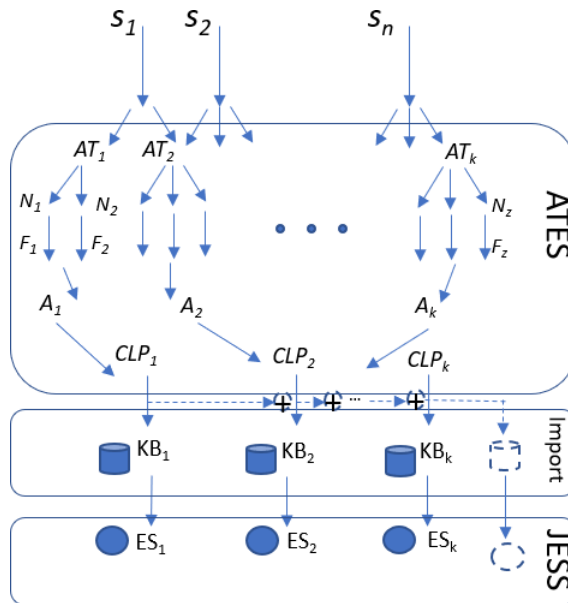
S4.1 pav. Metodo veiklos diagrama

Pasiūlytas metodas gali konvertuoti *UATS XML* formatu užrašytus atakos medžius. Jei turimas formatas yra kitoks, į *UATS XML* formatą galima konvertuoti panaudojant jau egzistuojantį *ATTop* įrankį. *ATES* įrankis yra mūsų *Python* pagrindu sukurta programa, skirta metodo praktiniam įgyvendinimui, ji atlieka *UATS* transformavimą į *JESS*. Transformuotos taisyklės išvedamos į *CLP* ir *DAT* failus (arba vieną sujungtą failą, jei naudojamas taip pasirinkta), kurie yra palaikomi *JESS* ekspertinės sistemos.

UATS transformavimo į *JESS* žinių bazės faktus metu kiekvienas atakos medžio elementas yra transformuojamas į atskirtą žinių bazės faktą. Pasiūlytas metodas yra universalus tuo požiūriu, kad sugeneruotos taisyklės gali būti panaudotos tiek atskiroje žinių bazėje, tiek sujungtos į vieną (S4.2 pav.).

Metodo eksperimentinio bandymo metu iš 22 skirtingų šaltinių ($S_1 \dots S_n$), kuriuos sudarė moksliniai straipsniai ir *ADTool* atakos medžių biblioteka, buvo surinkti skirtingi 49 atakos medžiai ($AT_1 \dots AT_k$). Atakos medžių šakos ir lapai ($N_1 \dots N_z$) transformuoti į faktus ($F_1 \dots F_z$), o atakos medžio viršūnės į atakos tikslus ($A_1 \dots A_k$). Kiekvienas atakos medis transformuotas į *CLIPS* taisykles, kurios išsaugotos naujai sugeneruotuose failuose

($CLP_1 \dots CLP_k$). Kiekvienas CLP failas saugomas kaip ekspertinės sistemos žinių bazė ($KB_1 \dots KB_k$), kuri gali būti importuojama į ekspertines sistemas ($ES_1 \dots ES_k$) arba sujungta į vieną žinių bazę. Eksperimentinio bandymo metu buvo gauti 875 ekspertinės sistemos žinių bazės faktai. Eksperimentų metu visi $UATS$ formato atakos medžiai buvo konvertuoti sėkmingai be klaidų.



S4.2 pav. Pasiūlyto metodo struktūros bendrasis vaizdas

Siekiant parodyti, kad automatiškai sugeneruotos žinių bazės taisyklės yra panaudojamos praktiškai, jos buvo importuotos į *JESS* pagrindų sukurtą ekspertinę sistemą ir atliktas rizikos vertinimas (S4.3 pav.).

*****	Technical Impact HIGH: 8.0	*****
*****	Business Impact MEDIUM: 5.0	*****
Probability of success:	0.9560546875	
Technical risk:	7.6484375	
Business risk:	4.7802734375	
Attack name:	Block communication	
Description:	How to block communication between an RFID reader and tag	

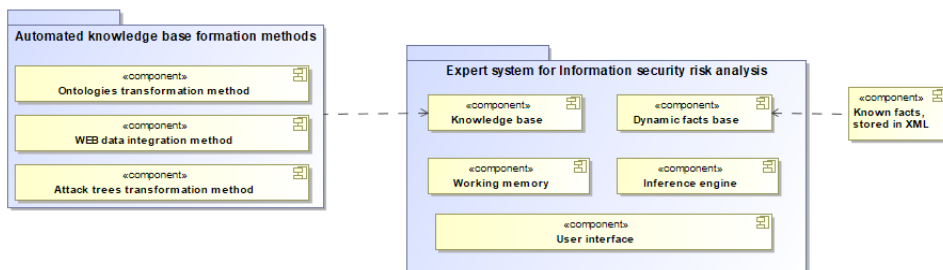
S4.3 pav. Rizikos vertinimo pavyzdys

Rizikos vertinimo pavyzdyje rizikos įvertis apskaičiuojamas pagal formulę $Rizika = Poveikis * Tikimybė$. Rizika apvesta raudonu stačiakampiu, poveikis žaliu, o tikimybė – mėlynu. Automatiškai sugeneruotos taisyklės taip pat buvo importuotos į 5 skyriuje aprašomą informacijos saugos rizikų analizės ekspertinės sistemos prototipą.

5. Informacijos saugos rizikų analizės ekspertinės sistemos, skirtos mažoms ir vidutinėms įmonėms, prototipas

Šiame skyriuje apžvelgiamas informacijos saugos rizikų analizės ekspertinės sistemos, skirtos smulkaus ir vidutinio dydžio įmonėms (*SME*), prototipas. Šią ekspertinę sistemą nuspręsta sukurti ir pritaikyti *SME* siekiant patvirtinti anksčiau pasiūlytus metodus. Kuriamos ekspertinės sistemos žinių bazė buvo užpildyta taisyklėmis, gautomis anksčiau pristatytų metodų eksperimentų metu.

Kuriamai ekspertinei sistemai pasirinktas *JESS* ekspertinių sistemų karkasas. Ekspertinės sistemos architektūra paremta tipine ekspertinės sistemos architektūra (S5.1 pav.).



S5.1 pav. Sukurtos ekspertinės sistemos architektūra

Skirtingai nei tipinė ekspertinė sistema, sukurta sistema naudoja dinaminę faktų bazę, kurioje išsaugomi jau iš ankstesnių vertinimų naudotojų pateikti faktai. Jeigu aplinkybės tarp vertinimų nekito, naudojami faktai iš faktų bazės, papildomai nebeužduodami susiję klausimai.

Sukurta ekspertinė sistema turi aprašytus šiuos penkis įmonių profilius:

1. Elektroninė komercija.
2. Programinės įrangos kūrimas.
3. Dizainas.
4. Transportas.
5. Medicina.

Dalis klausimų ekspertinėje sistemoje skirti tik konkrečiam įmonės profiliui, todėl naudotojui nereikia atsakyti į pagal įmonės profilį neaktualius klausimus. Ekspertinės sistemos žinių bazė buvo suformuota panaudojant taisykles, gautas anksčiau pasiūlytais metodais, taip pat ir rankines taisykles.

Siekiant patikrinti ekspertinės sistemos pritaikomumą praktikoje, buvo atlikti rizikų analizės eksperimentai. Eksperimentuose dalyvavo nepriklausomi 8 ekspertai. Ekspertų tinkamumas buvo patikrintas naudojant informacijos saugos srities klausimyną.

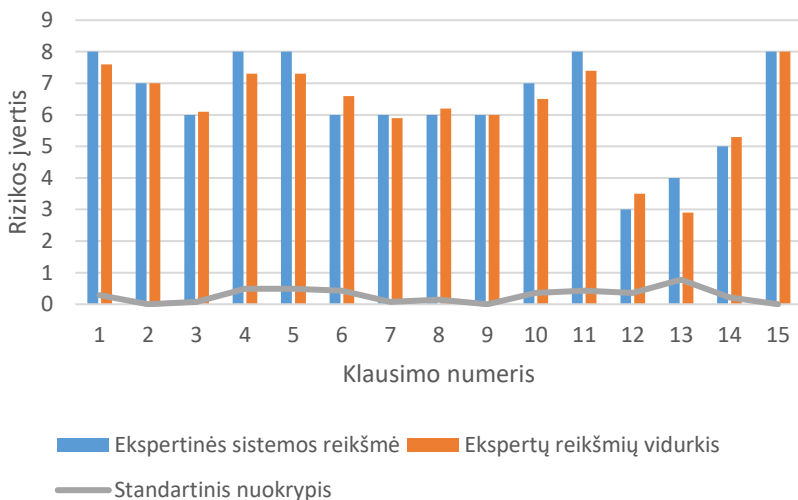
Eksperimento metu buvo atlikti dviejų scenarijų vertinimai su skirtingais įmonių profiliais, o S5.2 lentelėje pateikiami pagrindiniai eksperimente naudoti faktai.

S5.2 lentelė. Pagrindiniai eksperimento faktai

Faktai	1 scenarijus	2 scenarijus
Įmonės profilis	El. komercija	Programų kūrimas
Darbuotojų skaičius	5	80
Įrangos kiekis	20	10
Saugumo politika	Ne	Taip

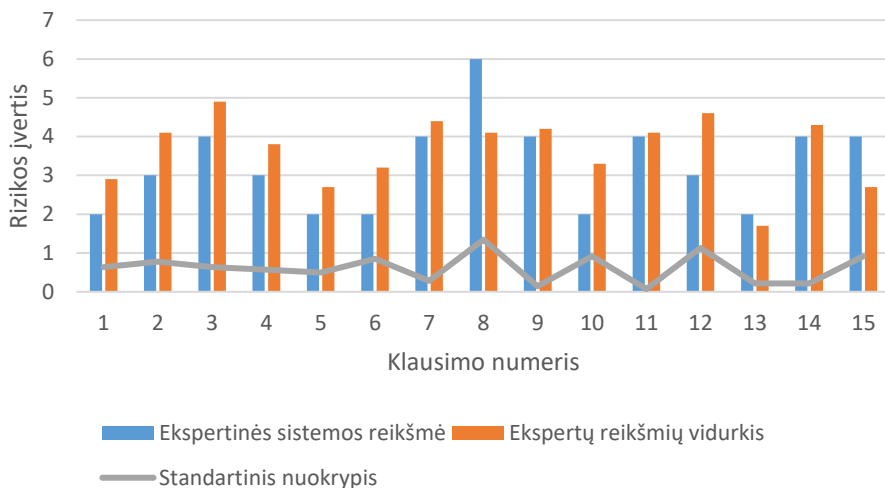
Eksperimento metu buvo 30 klausimų, po 15 kiekvienam scenarijui. Kiekvienas iš scenarijų turėjo po 1 klausimą įmonės profiliui nustatyti. Ekspertai turėjo įvertinti rizikas pagal pateiktus faktus: turto vertė, įdiegtos kontrolės priemonės, prieinama įranga ir kita informacija pagal įmonės profilį. Ekspertai taip pat turėjo pateikti vertinimų įverčius ir rekomendacijas rizikoms valdyti.

Eksperimento rezultatai parodė, kad ekspertinė sistema gali būti naudojama informacijos saugos rizikų analizei *SME* tipo įmonėse. Pirmo scenarijaus rezultatai rodo didelę ekspertų ir ekspertinės sistemos rezultatų koreliaciją (S5.3 pav.).



S5.3 pav. Pirmo scenarijaus rezultatai

Pirmo scenarijaus ekspertų ir ekspertinės sistemos rizikų įverčių koreliacijos koeficientas – 0,949. Antro scenarijaus ekspertų ir ekspertinės sistemos rizikų verčių koreliacija yra mažesnė, t. y. vidutinė (S5.4 pav.).



S5.4 pav. Antro scenarijaus rezultatai

Antro scenarijaus ekspertų ir ekspertinės sistemos rizikų įverčių koreliacijos koeficientas yra 0,588, t. y. koreliacija vidutinė. Toks įvertis yra pakankamas įrodyti ekspertinės sistemos praktinio taikymo galimybes. Vertinant abiejų scenarijų ekspertų ir ekspertinės sistemos rizikų įverčių koreliavimą, koeficientas yra 0,927, t. y. didelė koreliacija. Taigi, eksperimentas parodė, kad ekspertų ir ekspertinės sistemos gauti rezultatai koreliuoja.

Eksperimento metu buvo naudojamos automatiškai gautos taisyklės, todėl daroma išvada, kad anksčiau pasiūlytais metodais automatiškai gautos taisyklės gali būti naudojamos praktikoje.

Bendrosios išvados

1. Atlikta mokslinės literatūros apžvalga parodė informacijos saugos rizikos analizės ekspertinių sistemų žinių bazės pildymo automatizavimo poreikį, kadangi klasikiniai neautomatizuoti būdai yra per lėti ir brangūs. Dabartiniai moksliniai tyrimai šia tema koncentruoti į statinių duomenų, tokių kaip ontologijos, integravimą, tai nėra tinkama dažnai besikeičiančioje srityje, tokioje kaip kibernetinis saugumas. Atlikta mokslinės literatūros analizė leido patvirtinti, kad žinių bazės generavimo tobulinimas turėtų būti nukreiptas į egzistuojančių duomenų transformavimą į žinių bazės taisykles. Atlikus galimų šaltinių analizę, problemai spręsti pagal pasirinktus kriterijus, tokius kaip tinkamumas informacijos saugai, pritaikomumas rizikų vertinimui, esamų šaltinių pakankamumas, prieinamumas tekstiniu formatu ir tinkamumas automatinei transformacijai, buvo pasirinkti trys

žinių šaltiniai: ontologijos, žiniatinklio duomenys ir atakų medžiai. Atlikus ekspertinių sistemų karkasų analizę, pagal apibrėžtus kriterijus buvo pasirinktas *JESS* ekspertinės sistemos karkasas, kuris yra pakankamas šiame darbe iškeltiems uždaviniams įgyvendinti.

2. Šiame darbe buvo pasiūlyti ir įgyvendinti trys automatizuoti ekspertinių sistemų žinių bazės generavimo metodai, naudojant įvairius informacijos šaltinius: metodas, pagrįstas informacijos saugos ontologijų transformavimu į ekspertinių sistemų žinių bazės taisykles ir ją papildantis informacijos saugos kontrolės priemonių aprašymas; metodas, pagrįstas žiniatinklio duomenų, kuriuose yra kenkėjiško kodo aprašymai, transformavimu į informacijos saugos rizikų analizės ekspertinių sistemų žinių bazės taisykles ir ją papildantis dinamine informacija apie kenkėjiško kodo paplitimą pagal regionus; metodas, pagrįstas atakų medžių iš įvairių šaltinių transformavimu į žinių bazės taisykles, apimančias informacijos saugos rizikos vertinimui svarbią informaciją, pvz., grėsmės tikimybę, kainą, sudėtingumą, grėsmės išnaudojimo laiką ir atsakomąsias priemones.
3. Pasiūlytų metodų teisingumas užtikrinamas juos realizuojant, kad automatiškai generuojamos taisyklės atitiktų taisyklių sintaksės ir logikos reikalavimus, o veiksmingumas patvirtintas praktiniais eksperimentais, naudojant sukurtą ekspertinės sistemos prototipą.
 - 3.1. Metodų eksperimentinių bandymų metu į žinių bazės taisykles buvo transformuotos 2 praktinės ontologijos, kur automatiškai suformuotos taisyklės sudarė 60 proc. ir 64 proc. Iš viso buvo automatiškai sugeneruotos 662 taisyklės iš ontologijų. Žinių bazės formavimo metodu, pagrįstu žiniatinklio duomenų transformavimu, iš viso automatiškai suformuotas 201 žinių bazės įrašas, kuriame buvo renkami duomenys apie geografinių kenkėjiškų programų platinimą 3 šalyse: Lietuvoje, Latvijoje ir Estijoje. Atakų medžių transformavimo metu iš 22 skirtingų šaltinių surinkti 49 atakos medžiai, kuriuos transformavus į žinių bazės taisykles, gauta žinių bazė, kurią būtų galima panaudoti 49 kibernetinių atakų rizikai įvertinti ir pasiūlyti daugiau nei 826 galimus valdymo būdus.
 - 3.2. Ekspertinės sistemos žinių bazė buvo sukurta panaudojant automatiškai gautas ir rankiniu būdu gautas taisykles. Sėkmingas tokios žinių bazės suformavimas ir panaudojimas rodo, kad įvairiais būdais gautas taisykles galima integruoti į vieną žinių bazę. Tolesnis ekspertinės sistemos ir ekspertų atliktas palyginamasis rizikos vertinimas parodė praktinį ekspertinės sistemos pritaikomumą mažose ir vidutinio dydžio įmonėse, nes ekspertinės sistemos ir ekspertų rezultatai rodo didelį panašumą: koreliacijos koeficientai dviejuose scenarijuose buvo lygūs 0,949 ir 0,588 atitinkamai, o p reikšmės buvo mažesnės nei 0,05 ir lygios 3,36E-07 ir 0,039, todėl galima patvirtinti ekspertų ir ekspertų sistemos rezultatų panašumą.

Annexes¹

- Annex A.** Fixed Rules for Method of Transforming an Ontology into Rules
- Annex B.** Examples of the developed tools
- Annex C.** Expert Knowledge Verification Questionnaire
- Annex D.** Declaration of Academic Integrity
- Annex E.** Copies of Scientific Publications by the Author on the Topic of the Dissertation

¹ The annexes are supplied in the enclosed compact disc.

Donatas VITKUS

RESEARCH ON AUTOMATED KNOWLEDGE BASE GENERATION
METHODS OF INFORMATION SECURITY RISK
ANALYSIS EXPERT SYSTEMS

Doctoral Dissertation

Technological Sciences,
Informatics Engineering (T 007)

INFORMACIJOS SAUGOS RIZIKOS ANALIZĖS
EKSPERTINIŲ SISTEMŲ ŽINIŲ BAZĖS AUTOMATIZUOTO
GENERAVIMO METODŲ TYRIMAS

Daktaro disertacija

Technologijos mokslai,
informatikos inžinerija (T 007)

2022 04 26. 16,0 sp. l. Tiražas 20 egz.
Leidinio el. versija <https://doi.org/10.20334/2022-014-M>
Vilniaus Gedimino technikos universitetas
Saulėtekio al. 11, 10223 Vilnius
Spausdino BJ UAB „Baltijos kopija“,
Kareivių g. 13B, 09109 Vilnius