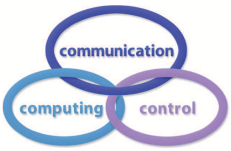
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(6), 743-758, December 2019.

Automated Expert System Knowledge Base Development Method for Information Security Risk Analysis

D. Vitkus, Z. Steckevicius, N. Goranin, D. Kalibatiene, A. Cenys



Donatas Vitkus*
Vilnius Gediminas Technical University, Vilnius, Lithuania
*Corresponding author: d.vitkus@vgtu.lt

Zilvinas Steckevicius
Vilnius Gediminas Technical University, Vilnius, Lithuania
zilvinas.steckevicius@vgtu.lt

Nikolaj Goranin
Vilnius Gediminas Technical University, Vilnius, Lithuania
nikolaj.goranin@vgtu.lt

Diana Kalibatiene
Vilnius Gediminas Technical University, Vilnius, Lithuania
diana.kalibatiene@vgtu.lt

Antanas Cenys
Vilnius Gediminas Technical University, Vilnius, Lithuania
antanas.cenys@vgtu.lt

Abstract: Information security risk analysis is a compulsory requirement both from the side of regulating documents and information security management decision making process. Some researchers propose using expert systems (ES) for process automation, but this approach requires the creation of a high-quality knowledge base. A knowledge base can be formed both from expert knowledge or information collected from other sources of information. The problem of such approach is that experts or good quality knowledge sources are expensive. In this paper we propose the problem solution by providing an automated ES knowledge base development method. The method proposed is novel since unlike other methods it does not integrate ontology directly but utilizes automated transformation of existing information security ontology elements into ES rules: The Web Ontology Rule Language (OWL RL) subset of ontology is segregated into Resource Description Framework (RDF) triplets, that are transformed into Rule Interchange Format (RIF); RIF rules are converted into Java Expert System Shell (JESS) knowledge base rules. The experiments performed have shown the principal method applicability. The created knowledge base was later verified by performing comparative risk analysis in a sample company.
Keywords: information security risk analysis, ontology, knowledge base, expert system, transformation, RIF, JESS.

1 Introduction

Many authors agree that today information is the critical business part despite the size of the enterprise. Companies are adapting to the time changes: growing speed of change and complexity, globalisation [16]. Even the smallest company has any information and information system, which are needed to be secured [35]. However, because of enterprises becoming more complex, integrated and connected to third parties, the security and controls budget quickly reaches its limitations [12].

The protection of information resources from the complex and rapidly evolving security threat landscape is a significant challenge to the modern organisation [46]. The reasons stated above motivate for searching of effective ways to increase information security level in organizations.

According to the importance of assets, it is necessity to analyse the potential risks to do not allow these risks to be converted into events [48]. As [10] said, risk management has proved to be efficient in the frame of the governance system due to its capacity to reduce costs associated with the management of different risks. According to Europe's General Data Protection Regulation, a risk-based approach to data protection is embraced [30]. So, all companies, including small and medium-sized enterprises (SME), have to protect their information systems. Enterprises have two ways of ensuring their information security: to employ a specialist or to outsource the risk analysis service. The problem is that the price of both choices is rather high.

Therefore, there is a need to automate the security risk analysis process by introducing expert systems or decision support systems, which could help enterprises to perform an information security risk analysis without any special knowledge and without hiring security experts and making security risk analysis process faster and cheaper [47]. There were some attempts to use expert systems for automating a particular part of security risk analysis [4, 5], cybersecurity incident prediction [42] and solving other real-worl problems also [29]. The main reason of such attempts is to make risk analysis process faster and cheaper [47]. Also, using expert systems helps to optimise asset management and their life cycle according to risk assessment, especially in specific sectors, like electric power transmission [41], power plant projecting [19]. However, it is necessary to mention that the development of an expert system knowledge base is an expensive and complicated process [24], [15]. So, the motivation of the paper is related to the need to automate the security risk analysis knowledge base development to minimize the expenses.

In this paper we propose a novel approach that allows the automatic transformation of existing information security ontologies into the expert system knowledge base rule set. Currently, a lot of information, including security information, like standards, best practices, is collected and presented in the ontologies [7]. Ontology comparing with usual not semantic database has particular advantages that can be used for developing expert systems [14]. Ontology can be expressed in several ways, for example, using graphical software, which makes ontology formation, maintenance, and usage easier [40]. Therefore, ontology presented in a particular software, could be used for its automatic transformation into the expert system knowledge base.

However, a deep understanding of ontology is needed, since any change in the ontology may require a change in the software's source code [27]. Nevertheless, these disadvantages cannot reproduce the expressiveness of ontology, and expert systems have their inference engine which is separate from the knowledge base. The main advantages of the work is that we provide a method for automated knowledge base development for an expert system, that can be used for risk analysis. It can help to provide an information security risk assessment for non-IT specialist cheaper and in a more effective way. The methods of information security risk analysis and the development of reasoning engines for expert systems are out of the scope of this paper.

The paper is organized as follows. Section 1, current section, is an introduction. Section 2

presents related works in security risk analysis and ontology usage for the development of expert systems knowledge base. Section 3 presents an approach of automatic transformation of existing security ontologies into the expert system knowledge base rule set and presents the implementation of the proposed approach of the information security standards ontology transformation into the JESS ES rules which is implemented in several steps with the help of a developed tool. OWL RL subset of ontology elements are segregated into RDF triplets, that are later transformed into RIF format. Then, RIF rules are converted into JESS knowledge bases rules with the help of an external tool. Conversion results are presented in Section 4. Finally, Section 5 concludes the paper.

2 Related work

Expert system is an computer-based system which has several parts. The basic architecture of an expert system is shown in Figure 1.

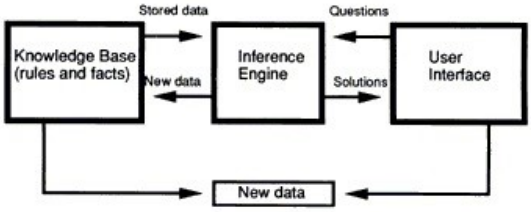


Figure 1: The basic architecture of an expert system [22]

The basic architecture of an expert system includes three main parts: Knowledge Base, Inference Engine and User Interface. One of the biggest problems in expert system usage is to get knowledge base updated [3]. Nevertheless, expert systems are used in nowadays for solving actual problems. For example, it can be successfully used as first-line in IT help-desk [13].

The majority of expert systems are developed using specialized software called a shell, especially when it comes to the rule-based expert systems. Shell allows the user just to create the knowledge base in the form of simple rules and not paying attention to knowledge interpretation engine development, thus minimizing the ES development time [2].

Nevertheless, forming the knowledge base is a complicated process that has a number of unsolved issues and illustrates the problems of this work. One of them is knowledge base integrity that is hard to maintain [24]. Adding new data can destroy existing rules, cause conflicts or endless cycles [25]. In result, it can make expert system work incorrectly.

There are some methods to solve knowledge base integrity problem. One of them is KADS (and its extension - CommonKADS) methodology [44]. The main idea is to design everything with UML diagrams [28]. The main disadvantage of that method is that it needs a lot of time resources, and the user can still make a mistake if a knowledge base is very complicated.

The use of information collected in the form of ontologies for ES knowledge base formation seems useful by many experts, but in practice, current research focus on ontologies and expert systems is separated. The main idea of ontology is to identify specific object classes and relations between them [8]. Ontology can be used in various business areas, but these areas can be divided into two parts: ontology usage as a vocabulary and ontology usage as a content [9].

Ontology usage as vocabulary enabled to develop the second generation of the web [18]. But ontology can be much more than a set of concepts or vocabulary - they have a lot of

applications in artificial intelligence. In this context, ontology can be used as information sources [23]. Ontology provides integrity and can be displayed by various graphical tools, has a common format and can be apprehensible in other systems, which designed to work with ontologies [31]. For example, XP.K (eXtreme Programming of Knowledge-based systems) methodology is based on Agile principles: to start from the simplest model of expert system and grow it with time [26]. Model should include just that what is really needed to solve existing problems. Knowledge base is created from ontology which is created by experts [38]. The main problem of this approach is that it requires a lot of time to develop the ontology from scratch on iteration basis and does not make use of already existing knowledge collected, while it could be valuable to use already existing information security-related ontologies.

Some attempts [34] for ontology use as a source for knowledge base formation, but the approach proposed destroys the idea of ES that knowledge base and inference engine should be separate. DAMLJessKB [21] software was developed to transform DAML (DARPA Agent Markup Language) ontology to the JESS expert system's rules, but it can work only with specific DAML ontology and JESS expert system. Another sample is DLEJena - the software, which transforms the OWL 2 RL profile, compatible with pD semantic, to Jena expert system platform [32]. The main disadvantage of these programs is that they transform only specific ontologies to the specific expert systems format.

On the other hand, there exist a number of information security ontologies that can be used in knowledge base formation of risk-analysis ES. Currently, most of them are used for other purposes. For example, ontology proposed in [39] can be used as a tool to identify the level of system vulnerabilities according to the internal users' accounts configuration and system configuration [33]. As the authors state, all illegal activity in the system is done by human resources and internal users have more privileges than external. This tool is based on taxonomy with users' settings and includes different behavioural motivation, for example, intentional and unintentional activity.

ROPE methodology and the related ontology is used for enterprise IT security evaluation, focuses on business processes and risk management [33]. 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, like ISO 27001/2 that can be used to align external standards with internal procedures [17]. One more security ontology is OntoSec [33] that can be used by security engineers can to set configuration effectively, according to the existing security events.

In [36] the 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 purposes. It was used for security standards mapping task and was linked with 4 security standards: ISO 27001, PCI DSS, ISSA 5173 and NISTIR 7621.

Summarasing it can be said that there are a lot of information security ontologies and our work allows transforming the selected ontology into expert system rules set in an automated and novel way.

3 Automated approach of ontology transformation into ES knowledge base

In order not to spend time resources for developing knowledge bases from scratch and to solve the limitations specific to earlier research in ontology transformation to knowledge base rule-sets we propose new method that allows transformation of existing information security ontologies into expert system rule-set via universal RIF format, that is later converted to the

format of a specific expert system. In fact, the method can be applied for any ontology type transformation to any ES knowledge base with some additional modifications, related to the syntax of ES rule defining language.

The method proposed is that it uses the RIF (Rule Interchange Format) format, which was presented by the W3C consortium in 2010. The primary purpose of this format is to transfer rules from one system to another [49]. Unlike other standards of Semantic Web (RDF, OWL, SPARQL), RIF was developed not for defining rules, but as a standard for rule transfer from one system to another, which became extremely important with the increase of standards for rule definition [43].

RIF has two dialects: based on logic and based on rules. Dialects based on logic include logical languages, like Horn logic and others. Dialects based on rules include productive rules. The same rules are used in expert systems [37]. RIF PRD (Production Rule Dialect) is a dialect based on production rules. Production rules semantic have format "IF condition THEN activity". Figure 2 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)
   )
 )
```

Figure 2: Example of RIF PRD

RIF was created in a way to be compatible with other W3C standards. That means that it can be compatible with RDF and OWL ontology languages [1]. Transferring these ontology languages into RIF can be done not only in the form rule-sets, but also in RDF triplets (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. In our experiments we were using OWL RL profile as a source. OWL RL rules can be divided into four categories (which may overlap):

Triplet 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 transformation.

Data type rules: type's comparison and verification. OWL and RIF supported types are transformed directly.

I.e. 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 the expert system. The method proposed in Figure 3 was implemented using `C#` language. During the first step, the tool developed converts the information security ontology into RIF standard. At the second step (RIF transforming into expert system production rules) the RIF PRD profile was used, which is 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.

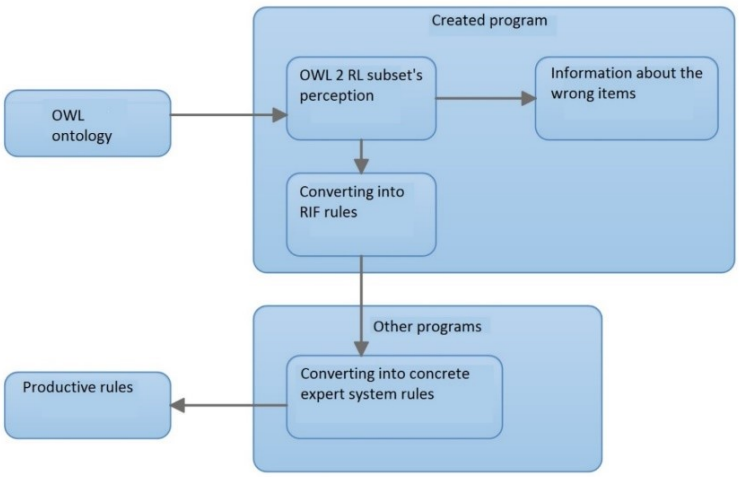


Figure 3: General view of the method proposed

The detailed method activity diagram is shown in Figure 4. The yellow border depicts actions that were proposed and implemented by the paper authors, while actions outside the yellow border are performed with the help of already existing methods and tools.

Model consists of primary data - ontology (OWL). Transformation of RIF rules is performed in several steps:

Scan ontology. OWL\XML ontology is scanned and checked, if the ontology is in OWL\XML format, since the developed tool supports only the most popular OWL\XML format. If ontology is stored in other formats, external tools should be used first to convert it into OWL\XML.

Check if the element is OWL RL. Only 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 non-OWL RL subset & Collect OWL RL subset. After successful scanning of 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 form of

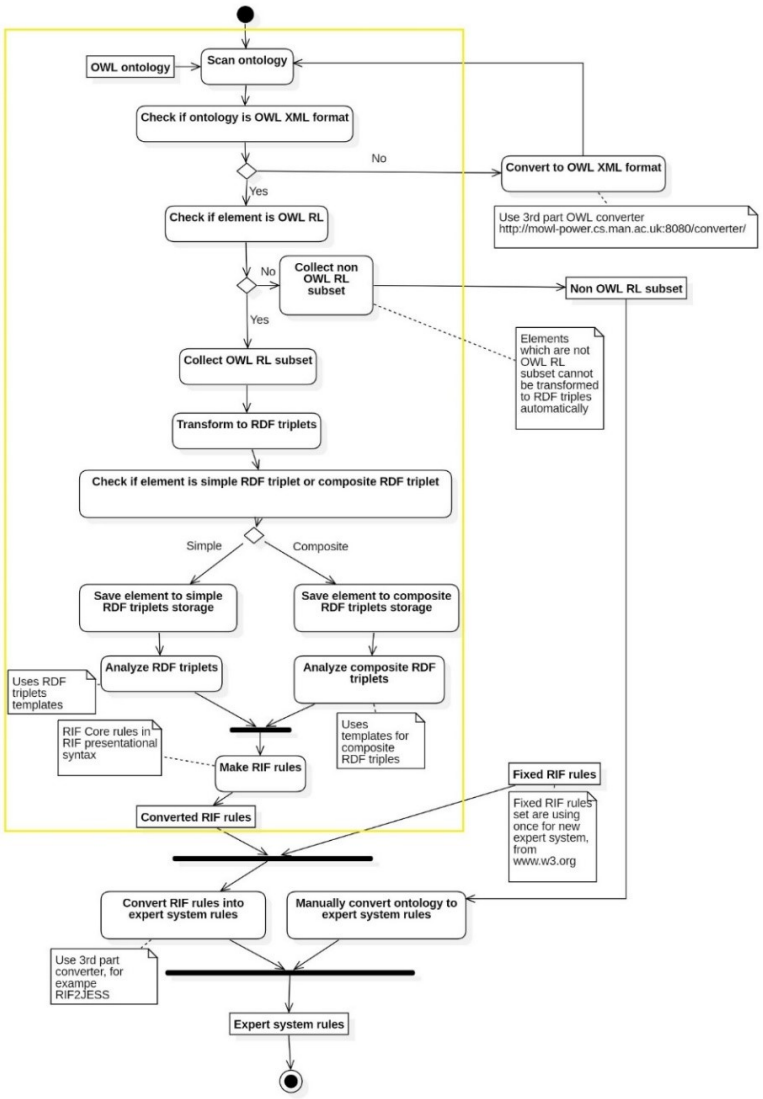


Figure 4: The detailed method activity diagram

RDF triplet format (subject, predicate, object). Check if the element is a simple RDF triplet or composite RDF triplet. Evaluation is performed if RDF triplet can be unlinked from other RDF triplets.

Save element to simple RDF triplet storage. RDF storage stores RDS triplets that can be unlinked from other RDF triplets, e.g. attribute type "symmetrical" is saved in the form of RDF triplet *(?p rdf:type owl:FunctionalProperty)*, where ?p is attribute name. Such kind of data is stored in the form of text registry, having 3 fields.

Save element to composite RDF triplet storage. Elements that cannot be unlinked are stored in a 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 3 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 the RDF triplet is predicate, which can acquire values from a finite set of possible

values because of ontology formality feature. The remaining two triplet elements - subject and object are specific concepts or attributes, defined in the ontology, that 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 formal algorithm in the form of pseudocode of OWL RL transformation into RIF rules is provided in Fig. 5

```
READ selected_owl_file;
WHILE NOT End_of_xml_tokens DO
   GET xml_nodes
END DO
FOREACH xml_node in document_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.
   ELSE DO collect_non_RL_elements //if XML node aren’t OWL 2 RL element, collect them separately
END DO

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

DO rdf_to_rif   //RDF triplets are converting to RIF rules, using patterns

PROCEDURE rdf_to_rif(IN rdf_triplets; OUT rule_1; OUT rule_2)
   READ rdf_triplet
      IF rdf_triplet_item2 == "rdf:type"
         THEN DO templateType(item, out rule1); //choose RIF pattern by the RDF triplet type
      IF rdf_triplet_item2 == "owl:equivalentProperty"
         THEN DO templateEquivalentProperty(item, out rule1, out rule2);
      //the similar for all RDF triplet 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[{0}->?y1] ?x[{0}->?y2]))", item.s1);
   //at all 20 similar RIF rules patterns
END
```

Figure 5: Pseudocode for OWL RL transformation into RIF rules

While analyzing XML nodes, RDF triplets are segregated. RDF triples are created according to the predefined 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 literal value. RDF triplets created in such a way are later transformed into RIF rules.

Group (
Forall $?v1_1$ ... $?v1_o$ *(*
$sr_1[pr_1->or_1]$ *:- And(* $s1[p_1->o_1]$ ... $sn[p_n->o_n]$ *))*
...
Forall $?vm_1$ ... $?vm_o$ *(*
$sr_m[pr_m->or_m]$ *:- And(* $s_1[p_1->o_1]$ ... $sn[p_n->o_n]$ *))*
)

where $?vi_1$ ... $?vi_o$ are the variables which occur in the rule.

Below the sample transformation is provided.

Two following RDF triplets are extracted from the ontology:

atributte_1 rdf:type owl:SymmetricProperty
atributte_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[atributte_1->?x] :- And(
     ?x[atributte_1->?y] ))
Forall ?x ?y (
 ?y[atributte_2->?x] :- And(
     ?x[atributte_2->?y] ))
```

The transformation process is composed of two parts: transformation of fixed rules and ontology rules, obtained from its T-Box axioms. In OWL 2 RL profile both ontology axioms and additional logical rules can be stored [45], but in this research, only the subset of axioms is used as a suitable source for ES knowledge base rules.

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

Another issue of ontology transformation - correctness of ontology that cannot be evaluated by the conversion program. Evaluation of ontology composion correctness if out of scope of this paper. 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 equation:

$R(RDF(O)) = Fixed\_rules \cup Ontology\_rules\ (RDF(O))$,

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

4 Results and discussion

The created program has five main parts: user interface; ontology scanning engine; analysis engine; ontology alignment to RDF triplet engine; transformation (into RIF rules) engine. The program class diagram is shown in Figure 6.

Program is composed of 6 main classes:

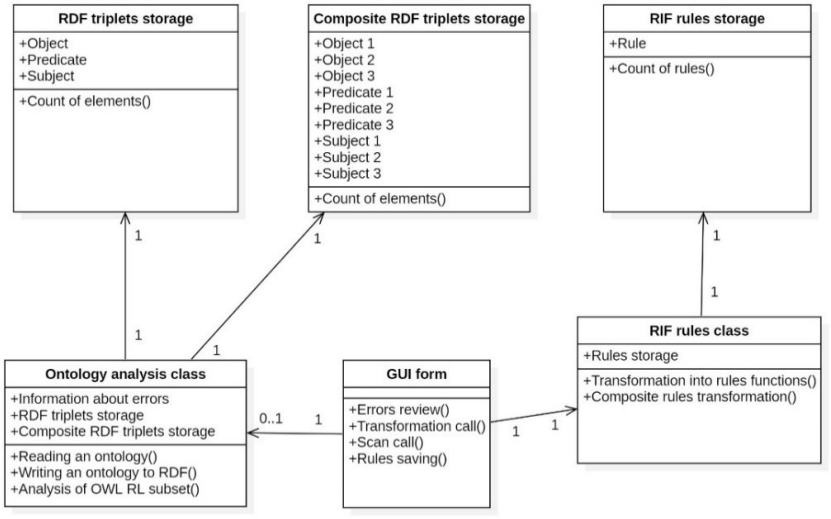


Figure 6: Program class diagram

GUI form. Program management. User can call ontology scanning, conversions into RIF rules, error review and rules saving functions.

Ontology analysis class. Responsible for 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 3 linked together complex RDF triplets that can not be unlinked.

RIF rules class. Converts RDF triplets into RIF rules as defined. **RIF rules storage.** Stores RIF rules in the form of presentational syntax that is later used for transformation into ES knowledge base rules.

All functionality is available from the main program window (Figure 7). The user has to press the "Scan ontology" button and specify the path to the ontology file.

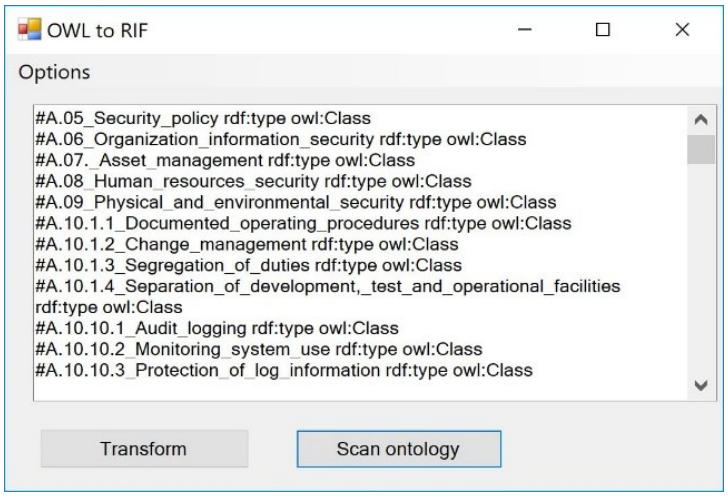


Figure 7: The program user interface

Ontology scanning engine imports (Figure 6) the ontology from the external .xml file of OWL\XML format. The program later performs the automatic scanning of elements, extraction

of OWL RL subset and performs conversion to RDF triplets. The results are shown in the main window. User, after reviewing the intermedium data and the list of errors, can press the "Transform" button. After that, RDF triplets are transformed to RIF rules and presented for user review. User can copy them or to save via program interface into the specified file.

User interaction is needed to initiate ontology imports and transformation processes. Program was tested with several ontologies: W3C consortium test ontologies [51], cloud security ontology [20] which is used in practice and information security standards ontology [36]. While W3C and Cloud security ontologies were developed by external parties, the Information security standards ontology was developed at Vilnius Gediminas Technical University in accordance with the formal OWL 2 RL rules defined in [11] and OWL 2 Full defined in [50]. Its presentation and in-depth description with verification was provided in [36].

All tested ontologies were relatively small (up to 1000 elements). The proportional size of the RL subset in ontologies used in tests was equal to 100% in case of W3C testing ontologies, and 65% - in case of Information security standards ontology. The proportion of the OWL RL subset in the case of Cloud security ontology was not analyzed. Execution time has not exceeded 50 ms (test platform: Intel Core i7-4710HQ 2.50 GHz, 8 CPU, 16GB RAM, Windows 8.1 OS). There were 15 ontologies at all. Results are shown in Table 1.

Table 1: Transformation results

Ontology	**The part of successfully transf. rules**
W3C testing ontologies, compatible OWL RL	90%
Cloud computing security ontology	64%
Information security standards ontology	60%

As can be seen from the transformation results, not all ontology elements were transformed. Some elements cannot be transformed into separate rules because they are just a part of other element or additional information. The best transformation result was achieved with test ontologies, that are based on the OWL RL profile. Specific security ontologies had shown worse results. In the case of the Information security standards ontology data types *xsd.real* and *xsd.rational* have caused 15% of unsuccessful conversion, 20% were caused by *SubClassOf* elements and 5% by other *xsd* data types. General method applicability was approved, since the transformation of even 60% can drastically decrease the knowledge base creation time.

For further verification purposes, the generated rules were integrated into the JESS-based prototype risk analysis ES, adapted for SMEs (enterprise size definition is adopted by EU recommendation 2003\361). This part of ES knowledge base was developed using traditional knowledge base development methods 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 mainly included information on appropriate security controls.

According to ISO 27001, information security risk is defined as a potential that some threat exploits an asset or assets group vulnerability and thus undermine the enterprise. Risks management process's purpose is to identify such risks, assess the likelihood of their occurrence and then take actions to reduce them to the acceptable level. Almost all risk analysis processes use the same method: identifying assets; identifying problems, threats, vulnerabilities; assessing risk likelihood and impact for assets [6]. After the risk analysis process, the user has to decide how to reduce risks; therefore, ES should not only evaluate risks, but also give some recommendations based on the acceptable risk level.

5 Conclusions and future work

The increasing demand for information security compliance and overall understanding of information security management importance leads a modern company to a need of a systematic risk management process. The use of expert systems for risk analysis is seen by many authors as a possible solution. Still, the development of expert systems is also a complicated and expensive task, and we indicate it as unsolved problem. Advantage of our method is that development of knowledge base can be partially simplified by using already available knowledge sources. Some earlier attempts by other authors were made to perform direct integration of security ontologies, that can be seen as a valuable source of information for expert systems knowledge base, with expert systems, but they lacked flexibility.

In this paper the new method was proposed that allows automatic transformation of existing information security ontologies into the expert system knowledge base ruleset. Information security standards ontology transformation into the JESS ES was implemented in several steps with the help of a developed tool: OWL RL subset of ontology elements are segregated into RDF triplets, that are later transformed into RIF format that can be easily converted into JESS knowledge bases rules with the help of external tool. The method supports the most popular OWL/XML format. Both simple (unlined) and complex (linked) OWL RL elements can be transformed. The biggest method advantage is that although ontology typically has a very sophisticated structure, it can be easily represented in the form of RDF triplets . The conversion test with different ontologies (W3C testing ontologies, "Cloud computing security ontology" and "Information security standards ontology" [36]) have shown 60-90% conversion success rate, that proves general method applicability, since automatic generation of even part of expert system knowledge base can decrease the general expert system development price. Verification of generated rules was performed by their integration with the knowledge base of a developed JESS-based expert system for risk analysis in SME. The obtained results have shown high correlation rate, but what is more important is that the generated rules were successfully integrated into the ES knowledge base.

Later research should be concentrated on tuning the conversion templates for achieving higher conversion success rates of ontologies and finding other sources for automatic filling of the ES knowledge base. Currently, we see the CVE (Common Vulnerabilities and Exposures) and attack trees as the next perspective sources for technical risk evaluation. Combination of these two structured sources should provide information on relevant threats as well as probabilities of different attack scenarios.

Funding

This research received no external funding.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Abbas, A.; Privat, G. (2018). Bridging Property Graphs and RDF for IoT Information Management, *SSWS@ ISWC*, 77–92, 2018.

[2] Abraham, A. (2005). *Rule-Based expert systems. Handbook of measuring system design*, John Wiley and Sons, New York, USA, 2005.

[3] Akerkar, R.A.; Sajja, P.S. (2010). *Knowledge-based systems*, Jones & Bartlett Publishers: Toronto, Canada, 2010.

[4] Atymtayeva L.; Kozhakhmet K.; Bortsova G. (2014). Building a Knowledge Base for Expert System in Information Security, *Soft Computing in Artificial Intelligence. Advances in Intelligent Systems and Computing*, Springer, Cham, 57-76, 2014.

[5] Benta, D.; Rusu, L.; Manolescu, M.J. (2017). Workflow Automation in a Risk Management Framework for Pavement Maintenance Projects, *International Journal of Computers Communications & Control*, 12(2), 155-165, 2017.

[6] Blackley, J.; Peltier. (2015). *Information Security Risk Analysis*, CRC Press: New York, USA, 2015.

[7] Blanco, C.; Lasheras, J.; Valencia-GarcĂa, R.; Fernandez-Medina, E.; Toval, A.; Piattini, M. (2008). A systematic review and comparison of security ontologies, *Availability, Reliability and Security, ARES 08. Third International Conference on IEEE*, 813-820, 2008.

[8] Bova, V.V.; Kureichik, V.V.; Lezhebokov, A. (2014). The integrated model of representation of problem-oriented knowledge in information systems, *2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT)*, 1-4, 2014.

[9] Brank, J.; Grobelnik, M.; Mladenic, D. (2005). A survey of ontology evaluation techniques, *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*, 166-170, 2005.

[10] Butaci, C.; Dzitac, S; Dzitac, I; Bologa, G. (2017). Prudent decisions to estimate the risk of loss in insurance, *Technological and Economic Development of Economy*, 23(2), 428-440, 2017.

[11] Cao, S. T.; Nguyen, L. A.; Szalas, A. (2011). On the Web ontology rule language OWL 2 RL, *International Conference on Computational Collective Intelligence*, Springer, 254-264, 2011.

[12] Classically, I. (2010). Performing a Security Risk Assessment, *ISACA Journal*, 1, 1-7, 2010.

[13] Dahouk, A. W.; Abu-Naser, S. S. (2018). A Proposed Knowledge Based System for Desktop PC Troubleshooting, *International Journal of Academic Pedagogical Research*, 2(6), 1-8, 2018.

[14] 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, 2016.

[15] de Rosa, F.; De Gloria, A.; Jousselme, A. L. (2019). Analytical games for knowledge engineering of expert systems in support to Situational Awareness: The Reliability Game case study, *Expert Systems with Applications*, 138, 112800, 2019.

[16] Dzitac, I.; Barbat, B. E. (2009). Artificial intelligence+ distributed systems= agents, *International Journal of Computers Communications & Control*, 4(1), 17-26, 2009.

[17] Fenz, S.; Plieschnegger, S.; Hobel, H. (2016). Mapping information security standard ISO 27002 to an ontological structure, *Information & Computer Security* , 24(5), 452-473, 2016.

[18] Gruber, T. (2008). Collective knowledge systems: Where the social web meets the semantic web, *Web semantics: science, services and agents on the World Wide Web*, 6(1), 4-13, 2018.

[19] Islam, M. S.; Nepal, M. P.; Skitmore, M.; Kabir, G. (2019). A Knowledge-based Expert System to Assess Power Plant Project Cost Overrun Risks, *Expert Systems with Applications*, 138, 12-32, 2019.

[20] Janulevicius, J.; Marozas, L.; Cenys, 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 (eStream)*, IEEE, 1-6, 2017.

[21] Joseph, K.; William, R. (2003). DAMLJessKB: A Tool for Reasoning with the Semantic Web, *IEEE Intelligent Systems*, 18, 74-77, 2003.

[22] Kaminski, J. (2014). *Nursing Decision Support and Expert Systems & Artificial Intelligence*, [Online]. Available: http://www.nursing-informatics.com/, Accessed on 03 March 2019.

[23] Katz, Y.; Grau, B.C. (2005). Representing qualitative spatial information in OWL DL, *Proceedings of the First International Workshop: OWL Experiences and Directions*, Galway, Ireland, 2005.

[24] Kidd, A. (1987). Knowledge Acquisition - An Introductory Framework, *Knowledge acquisition for expert systems: A practical handbook*, Plenum Press: New York, USA, 5 - 15, 1987.

[25] 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, *Current Issues in Knowledge Management*, IGI Global: Paris, France, 228-242, 2008.

[26] Knublauch H. (2002). *A method for knowledge modeling with unified modeling language (UML): Building a blueprint for knowledge management*, PhD thesis, University of Ulm, 2002.

[27] Kontopoulos, E.; Martinopoulos, G.; Lazarou, D.; Bassiliades (2016). An ontology-based decision support tool for optimizing domestic solar hot water system selection, *Journal of Cleaner Production*, 112, 4636-4646, 2016.

[28] Kozhakhmet, K.; Bortsova, G.; Inoue, A.; Atymtayeva, L. (2016). Expert System for Security Audit Using Fuzzy Logic, *Proceedings of the 23rd Midwest Artificial Intelligence and Cognitive Science Conference (MAICS2012)*, 146-151, 2016.

[29] Ma, X.; Zhan, J.; Ali, M. I.; Mehmood, N. (2018). A survey of decision making methods based on two classes of hybrid soft set models, *Artificial Intelligence Review*, 49(4), 511-529, 2018.

[30] Maldoff G. (2017). *The Risk-Based Approach in the GDPR: Interpretation and Implications*, [Online]. Available: https://iapp.org/media/pdf/resource_center/GDPR_Study_Maldoff.pdf, Accessed on 03 March 2019.

[31] Mas, S.; Wang, F.; Reinhardt, W. (2005). Using ontologies for integrity constraint definition, *Proceedings of the 4th international symposium on spatial data quality*, 25-26, 2005.

[32] Meditskos, G.; Bassiliades, N. (2008). Combining a DL Reasoner and a Rule Engine for Improving Entailment-Based OWL Reasoning, *International Semantic Web Conference*, Karlsruhe, Germany, 277-292, 2008.

[33] Obrst, L.; Chase, P. (2012). On Developing on Ontology of the Cyber Security Domain, *Proc. of the 7-th International Conference on Semantic Technologies for Intelligence, Defense and Security*, Fairfax, USA, 49-56, 2012.

[34] Otero-Cerdeira, L.; Rodriguez-Martinez, F.J.; Gomez-Rodriguez, A. (2015). Ontology matching: A literature review, *Expert Systems with Applications* , 42.2, 949-971, 2015.

[35] Rainer, R.K.; Cegielski, C.G.; Splettstoesser-Hogeterp, I.; Sanchez-Rodriguez, C. (2014). Information Systems within the Organization, *Introduction to information systems. Supporting and Transforming Business*, 3rd ed., John Wiley & Sons: Toronto, Canada, 2014; 227-228, 2014.

[36] Ramanauskaite, S.; Olifer, D.; Goranin, N.; Cenys, A. (2013). Security ontology for adaptive mapping of security standards, *International Journal of Computers Communications & Control*, 8(6), 878-890, 2013.

[37] Reynolds, D. (2010). OWL 2 RL in RIF, *W3C Working Group Note*.

[38] Rick, U.; Vossen, R.; Richert, A.; Henning, K. (2010). Designing agile processes in information management, *2010 2nd IEEE International Conference on Information Management and Engineering*, 156-160, 2010.

[39] Sicilia, M.A.; Garcia-Barriocanal, E.; Bermejo-Higuera, J.; Sanchez-Alonso, S. (2015). What are information security ontologies useful for?, *Research Conference on Metadata and Semantics Research*, Springer, Cham, 51-61, 2015.

[40] Slimani, T. (2015). Ontology development: A comparing study on tools, languages and formalisms, *Indian Journal of Science and Technology*, 8(24), 1-12, 2015.

[41] Spatti, D. H., Liboni, L., Flauzino, R. A., Bossolan, R. P., Vitti, B. C. (2019). Expert System for an Optimized Asset Management in Electric Power Transmission Systems, *Journal of Control, Automation and Electrical Systems*, 30(3), 434-440, 2019.

[42] Sun, N.; Zhang, J.; Rimba, P.; Gao, S.; Zhang, L. Y.; Xiang, Y. (2018). Data-driven cybersecurity incident prediction: A survey, *IEEE Communications Surveys & Tutorials*, 21(2), 1744-1772, 2018.

[43] Tommasini, R.; Bonte, P.; Della Valle, E.; Ongenae, F.; De Turck, F. (2018). A Query Model for Ontology-Based Event Processing over RDF Streams, *European Knowledge Acquisition Workshop*, Springer, Cham, 439-453, 2018.

[44] Tsudik, G.; Summers, R. C. (1990). AudES-An Expert System for Security Auditing, *IAAI*, 221-232, 1990.

[45] Van Woensel, W.; Abidi, S. S. R. (2018). Optimizing semantic reasoning on memory-constrained platforms using the RETE algorithm, *European Semantic Web Conference*, Springer, Cham, 682-696, 2018.

[46] Webb, J.; Ahmad, A.; Maynard, S.B.; Shanks, G. (2014). A situation awareness model for information security risk management, *Computers & security*, 44, 1-15, 2014.

[47] Willcocks, L. (1994). *Information management: the evaluation of information systems investments*, 1st ed. Springer Science & Business Media: Oxford University, United Kingdom, 219-225, 1994.

[48] Yazdani, M.; Alidoosti, A.; Zavadskas, E.K (2011). Risk analysis of critical infrastructures using fuzzy COPRAS, *Economic research-Ekonomska istrazivanja*, 24(4), 27-40, 2011.

[49] Yu, L. (2011). *A developer's guide to the semantic Web*, Springer Science & Business Media: Oxford University, United Kingdom, 2011.

[50] *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)*, [Online]. Available: https://www.w3.org/TR/owl2-syntax/, Accessed on 03 March 2019.

[51] *W3C OWL 2 Web Ontology Language Conformance (Second Edition)*, [Online]. Available online: https://www.w3.org/TR/owl2-conformance/#Test_Cases, Accessed on 03 March 2019.