Theoretical and Applied Informatics
ISSN 1896–5334
Vol. 25 (2013), no. 2
pp. 69–91
DOI: 10.2478/thai-2013-0006

Ontology Based Model of the Common Criteria Evaluation Evidences

Andrzej Białas

Institute of Innovative Technologies EMAG
ul. Leopolda 31, 40-189 Katowice, Poland
a.bialas@emag.pl

Received 3 April 2013, Revised 27 August 2013, Accepted 26 Sptember 2013

Abstract: The paper presents a new ontology-based approach to the elaboration and management of evidences prepared by developers for the IT security evaluation process according to the Common Criteria standard. The evidences concern the claimed EAL (Evaluation Assurance Level) for a developed IT product or system, called TOE (Target of Evaluation), and depend on the TOE features and its development environment. Evidences should be prepared for the broad range of IT products and systems requiring assurance. The selected issues concerning the author's elaborated ontology are discussed, such as: ontology domain and scope definition, identification of terms within the domain, identification of the hierarchy of classes and their properties, creation of instances, and an ontology validation process. This work is aimed at the development of a prototype of a knowledge base representing patterns for evidences.

Keywords: Common Criteria, IT security evaluation, knowledge engineering, modelling, ontology, assurance methods.

1. Introduction

The paper deals with the improvements of the IT product or system development compliant with the ISO/IEC 15408 Common Criteria (CC) methodology [1], presenting a new ontological approach to the elaboration of evidences that ought to be developed and provided for an IT security evaluation process. The work exemplifies how to apply knowledge engineering methodology to the security engineering domain. The aim of the researches is to produce evidences better structured and more precise and inherent the Common Criteria requirements.

The CC methodology concerns IT products (hardware, software, firmware) and systems, which are called together TOEs (*Target of Evaluation*), because their security can be evaluated. After development and evaluation these IT products and systems are embedded in their managed operational environments, encompassing: other co-operating IT products or systems (evaluated or not) and their information resources. Within this environments different kinds of users and intruders are considered, as well as physical protection- and security administration issues.

Today's IT applications, especially those used in large businesses, industry, e-government and e-health sectors require dependable IT solutions. The Common Criteria methodology plays the key role in providing assurance for these IT products or systems. The assurance [1] is related to the confidence that an entity, i.e. IT product or system (TOE), meets its specified security objectives. The rigour applied to the IT security development process influences the assurance, and the results of this process are independently evaluated according to the standard. The measures were defined for the assurance, i.e. the EAL scale (*Evaluation Assurance Level*) in the range from EAL1 (min) to EAL7 (max). The TOE developers should provide evidences that the TOE meets the claimed EAL requirements described in Part 3 of the standard [1]. There is a broad range of IT products and systems working in risky environments, and thus requiring assurance.

The elaboration of evidences for the given TOE should reflect the rigour derived from the declared EAL, as well as the features of the IT product or system, character of the development, manufacturing and operational environments.

Going from the particular requirements contained within the components to the structured documents presenting evidences is not easy and requires common understanding of terms, mastering many interrelated details, specialised know-how and generally – knowledge and experiences. Most of the related works are usually unpublished. IT developers, users, stakeholders, evaluators, and managers find it hard to master the CC compliant IT development and evaluation processes, which are difficult and expensive [2]. If the processes are improved, the existing barriers in the dissemination of higher assurance IT products or systems will be decreased.

The objective of the presented works is to improve the elaboration of evidences provided for the evaluation process, using advantages and new possibilities brought by the ontological approach. The improvement generally concerns: better formalization and preciseness, software support, and domain knowledge management of CC compliant IT development processes.

The paper includes three main sections. Section 2 introduces basic knowledge engineering issues and reviews concerning the application of ontologies to protect information security, placing the author's works in this field. Section 3 characterizes the domain of the discussed ontology, i.e. CC compliant IT security development- and foremost TOE development processes. Section 4 presents an ontology elaboration process using basic knowledge engineering principles and focusing on the issues dealing with evidences.

2. Ontological approach in information security domain

The term ontology was adopted for computer science from philosophy. In artificial intelligence literature there exist a few definitions of this term. Generally, an ontology represents explicit formal specifications of a set of concepts within a domain of knowledge and the relationships between those concepts [3]. Concepts are identified with classes. A class can be considered a set of similar primitives called instances. The class may have subclasses, which are more specific than the class itself (superclass). The concepts may have different attributes or features called properties. The properties may have restrictions assigned. In this paper, the following terms will be used: class, instance (older equivalent term – "individual" will be used in the ontology editor only), properties and restrictions. These terms will be applied here with the context of the domain of discourse (domain of knowledge) that can be defined as "Common Criteria compliant IT security development and evaluation". This domain can be considered a subdomain of a broader one – a security engineering or information security domain.

An ontology together with a set of individuals of classes, properties and restrictions constitutes a knowledge base. In this context, the ontology can be viewed as the data model of a data base, similar to hierarchical and relational models, but designed there for modeling knowledge. The ontology may be also used for reasoning, i.e. "looking for reasons for beliefs and conclusions" within a knowledge domain. Ontologies are specified with the use of languages, like OWL (*Web Ontology Language*) from World Wide Web Consortium (W3C).

Ontologies are elaborated to achieve the following aims [3]:

- to share common understanding of the structure of information among people or software agents,
- to support knowledge reusability,
- to make domain assumptions explicit,
- to isolate the domain knowledge from operational knowledge,
- to facilitate domain knowledge analyses.

It will be shown in the paper that all these aims are relevant to the ontology presented here and the related knowledge base.

Ontologies were elaborated recently in many disciplines where "common understanding", "common taxonomy" or "reasoning" are important, such as: artificial intelligence, Semantic Web, medicine, public administration, systems engineering, software engineering, biology, biomedical informatics, library science, and discussed here security engineering.

Analysing the current state of the art, some ontologies directly related to the Common Criteria methodology and some representing common security issues are encountered. Both ought to be briefly reviewed here.

While reviewing the CC-related ontologies, the following two works can be considered the most relevant:

- the work [4] presents an ontological approach to the modelling of the CC security functional requirements ([1]/Part 2) and their mapping to the specified security objectives with the use of the elaborated CC ontology tool called GenOM; the work focuses neither on other stages of the security target workout, like: security problem definition and elaboration of security functions, nor on the workout of evidences;
- the work [5], related to the ontology [6], is focused on the ontological representation of CC assurance requirements ([1]/Part 3) and presents a tool which supports evaluators during the certification process in such activities like: planning an evaluation process, reviewing relevant documents or creating reports; this tool allows a query of the data structure using RDF-based (*Resource Description Framework*) or OWL-based query languages, such as SPARQL (*SPARQL Protocol And RDF Query Language*); generally, this tool reduces the time and cost of the (evaluation) certification process; please note that the methodology focuses on the evaluation of assurance requirements, neither on the evidences elaboration, nor on the security target/protection profile workout.

Apart from the ontologies related directly to the CC methodology, other information security ontologies were elaborated. They do not concern the CC evidences issue but can be used as auxiliary ontologies during IT security development. The work [7] discusses security and trust ontologies, expressing risk analysis issues, security algorithm taxonomy, security functions, attacks and defence, and trust. In the work [8] the authors specify the extensive NRL (*Naval Research Laboratory*) security ontology, encompassing subontologies concerning the security of services, security agents, information objects, security algorithms, assurance and credentials. Additionally, they discuss ontology integration issues. The work [9] deals with the ISO/IEC 27001 standard implementation, [10] – general aspects of security management, [11] – quantitative risk analysis, [12] – selection of controls, and, finally, the work [13] – incident management issues. The examples of common security issues ontologies [14-15], [16], [17] can be analysed with the use of the Protégé Ontology Editor and Knowledge Acquisition System [18]. Please note the latter one [17] presenting the REI ontology (exactly: the set of subontologies) used for the security policy development. This issue has certain similarities to the evidences elaboration issue.

The review shows that the basic information security areas are represented by ontologies. They provide a unified set of terms and relationships in the particular domain and are comprehensible both to software agents and people. They are developed independently, which may cause incompatibilities. None of them encompasses the entire IT security development process in a complex way and none of them considers composing and management of the required Common Criteria evidences.

3. Common Criteria compliant IT security development- and evaluation processes as the domain of knowledge

The considered domain of knowledge encompasses three CC-related processes, but currently only the first two are covered by the discussed here Specification Means Ontology (SMO):

- IT security development process, related to the security target elaboration – specifying the TOE security functions which meet security requirements;
- TOE development process, related to the elaboration of an IT product or system and its documentation (including evidences) – implementation of these security functions at the claimed EAL; the paper is focused on this issue;
- IT security evaluation and certification performed by an independent body – assessment of the ST and the TOE against security assurance criteria in order to answer if EAL is met.

The paper [19] presents an ontological model of evidences related to the IT security development process only exemplified on the methane detector in mines. However, the paper discusses the ontological model of evidences related to the TOE development process, so the IT security development process will be presented here very briefly, as a background.

The IT security development process, related to the *security target* (ST) elaboration directly on the users' requirement, includes:

1. Preparing the ST introduction which contains different identifiers and informal descriptions of the TOE;
2. Security problem defining (SPD); SPD specifies threats, OSPs (organizational security policies) and assumptions;
3. Solving this problem by specifying security objectives (SO) – for the TOE and its development – and operational environments;
4. Working out the security functional requirements (SFRs) specification on the security objectives basis and a set of security assurance requirements (SARs)

which are derived mainly from the declared EAL (please note: EALs are predefined packages of SARs);

5. Preparing the TOE summary specification (TSS), containing the security functions (SF) derived from the SFRs that should be implemented in the IT product or system during the next step – TOE development process.

During the technology dependent TOE development process the security functions are implemented within the TOE, according to the rigour and details implied by SARs (components) of the declared EAL. Components are grouped by families, and families by CC classes, describing the following issues:

- TOE architecture, functional specification, design, implementation and security policy – expressed by the ADV (*Development*) class components,
- configuration management, life cycle, product delivery, development process security, used tools, flaw remediation – represented by the ALC (*Life cycle support*) class components,
- tests specification, test depth and coverage – implied by the ATE (*Tests*) class components,
- product manuals and procedures, worked out according to the AGD (*Guidance documents*) class components,
- vulnerability assessment according to the AVA (*Vulnerability Assessment*) class components.

The TOE development process provides evidences confirming that the TOE meets its EAL for the IT security evaluation process (not discussed here). The evidences, derived from SARs included in the EAL, encompass different kinds of documents, e.g.: user and technical documentation, tests, procedures, reports from analyses, documented behaviour, system records, etc. Some evidences concern directly the TOE while the others its development-, manufacturing- or operational environments. They are iteratively elaborated by different actors, mostly by developers, but there exist some kinds of evidences (independent testing, vulnerability analyses) provided by evaluators.

The presented ontological models of the SMO make use of the results of the author's earlier works [20] and the monograph [21] presenting the Common Criteria compliant, UML/OCL-based IT security development framework (ITSDF), which encompasses:

- models of the data structures and processes of IT security development stages, including: security problem definition, security objectives elaboration, security requirements, and, finally, security functions workout;

- models of the specification means used for these IT security development stages, including not only CC components but also the introduced semiformal generics, called “enhanced generics”; enhanced generics [21], derived from commonly used “generics”, are defined there as mnemonic names expressing common features, behaviours or actions related to IT security, like: subjects, objects, threats, assumptions, security policies, security objectives or functions; they are “enhanced” because they are semiformal and have features comparable to CC components, allowing parameterization, derivation, iteration, refinement, etc.

The semiformal ITSDF framework was implemented as a software tool aiding IT security developers. These works do not concern the elaboration of evidences in details. Evidences are considered there as documents that can be attached to a project and mapped to SARs as a whole.

UML-based models of enhanced generics and CC components included in the ITSDF framework were used to develop models contained in the Specification Means Ontology. Moreover, SMO was provided with models concerning evidences – discussed here.

4. Extending the Specification Means Ontology by the CC-related evaluation evidences

The work related to the Common Criteria processes improvement by applying the knowledge engineering methodology is extensive and for this reason was divided into a few parts. Initially, the Security Target Ontology (STO) [22], including the concepts related to the structures of the *security target* (ST), *protection profile* (PP) and their low assurance versions is elaborated. It does not provide, however, the specification means to fulfil the ST/PP structures with “contents”, specific for the given IT product or system. This role is taken over by the Specification Means Ontology [23-26].

The Specification Means Ontology, compliant with CC v. 3.1, embraces specification means used in the IT security development process, i.e.:

- author’s defined enhanced generics for assets, subjects, threats, OSPs, assumptions and security functions specifications,
- CC-defined functional and assurance components for security functional requirements (SFRs) and security assurance requirements (SARs) specification.

Here it is discussed how to extend this SMO ontology by the issues related to the TOE development process, i.e. evaluation evidences. This extension can be considered a separate ontology development process, embracing all typical activities of this process.

A general introduction concerning ontologies was placed earlier in Section 2, while the entire process of the ontology and knowledge base elaboration will be shown here. The paper is focused on the TOE development process as part of the SMO knowledge domain.

SMO elaboration (extension) is performed according to the basic knowledge engineering rules [3] and with the use of the Protégé Ontology Editor and Knowledge Acquisition System developed at Stanford University [18]. SMO will be expressed by the OWL language, precisely OWL-DL (*DL – Description Logics*) which allows automatic reasoning. The ontology development is generally an iterative and top-down process.

The SMO development process was validated with the use of the Protégé tool on some projects, including a simple firewall system (*MyFirewall*). The *MyFirewall* project, described in details in Appendix E of the monograph [21], was developed on the basis of "Annex D Worked Example: Firewall PP and ST" [27]. Using this UML/OCL *MyFirewall* project version, ontological models were built and later validated [23-24] in the range of the IT security development process only. Using the security functions specified there the paper extends this project example to the TOE development process and related evidences work-out.

The following subsections describe the ontology development process according to the basic knowledge engineering rules [3], [18].

4.1. The domain and scope of the ontology and competency questions

The domain of the SMO ontology is a Common Criteria compliant IT security development process, with its enhanced generics and components [23-24] items and here discussed TOE development process, with evidence items. The SMO domain can consist of two subdomains related to these processes.

SMO provides common taxonomy for all above mentioned items, allowing to better understand them and relationships between them. On the SMO basis a knowledge base has been developed which allows to retrieve right specification means for any IT security development stage and proper evidence patterns to compose evidences for the given IT product or system and the EAL declared for them. TOE developers can issue queries into the SMO related knowledge base, sampling information helping to answer different questions, such as: "How to express the considered issue, e.g. non-repudiation, by the right security objectives?", "Which security objectives can be selected to counter a given threat or enforce a given OSP?", "How to manage the configuration of the TOE on EAL4?", "How to perform a test coverage analysis for EAL3?", "What is the evidence pattern to elaborate EAL3 evidences concerning the test coverage?", "How to find information on the configuration management scope for the TOE evaluated against EAL5?", "What are predefined security functions concerning, for example, signature-based intrusion detection?". The questions that the ontology related knowledge base is able to answer are called competency questions [3]. The answers define the scope of the ontology. Please note that ontologies and their knowledge bases are developed incrementally, and after exceeding "a critical mass" a knowledge base allows to get answers to more and more advanced questions. The SMO has been currently extended by evaluation evidences.

4.2. The domain and scope of the ontology and competency questions

The next step of the ontology development [3] aims at reusing certain existing ontologies. The key issues are the range of compatibility, integration ability, quality, satisfied needs of the ontology users and, first and most – the availability of the given ontology. It is possible to use the third party developed CC components ontology, e.g. [4-5] instead of one's own developed ontology. However it seems to be unnecessary because ontologies described in the above mentioned works have features comparable to SMO, and those ontologies omit evidences. The straightforward SMO integration with common security issues ontologies, representing assets, threats, vulnerabilities, countermeasures, etc., is not easy due to their incompatibility with the CC methodology. Still, they can be helpful as an auxiliary source of knowledge, e.g. to define new enhanced generics. During the SMO development some experiments with the import of the [16] ontologies were performed but at this SMO ontology development stage it is not a key issue.

It should be emphasised that there is no ontology expressing Common Criteria evidences that can be reused in SMO.

The core ontology for the project is the STO ontology [22] which represents terms and relationships in the IT security development process. It will use the developed SMO ontology, providing predefined specification means and evidence patterns for this process. The STO and SMO ontologies are being integrated currently, but this integration is not a subject of the paper.

The new possibilities in this area can appear because different ontologies are still developed. It is necessary to investigate the ontology reusability issue so that it could better meet the needs and expectations of IT security developers.

4.3. Identifying important terms in the ontology

This issue should be considered separately for both SMO subdomains, related to the IT security development and TOE development processes.

The "identification of important terms (concepts)" for the first subdomain was performed mostly during the ITSDF framework elaboration and during the author's application works. The analyses were performed of the IT security development process, functional and assurance components [1], previously evaluated products and systems, case studies, etc.

The identification of terms related to the second subdomain is based on the analysis of security assurance components, different application notes [1] (Part 3/Appendix A), BSI guide [28] and the author's experiences from commercial projects.

4.4. The classes and class hierarchy

The previously identified terms and relationships should be expressed in a more formal way, i.e. as classes, their instances and properties.

First, the classes and their hierarchy are elaborated, related to the taxonomy of terms expressed by the ontology. Different analyses of terms and relations between terms should be performed and the terms should be ordered, e.g.: class-instance, class-subclass, class-superclass. Some classes are abstract and some have instances. It is also important to decide what is to be expressed by a class and what by a property. Please note that many correct solutions may exist, depending on the ontology developers' approach. Different factors ought to be taken into account, like: the possibility of the future evolution of the class hierarchy or integration with other ontologies, transitivity of class relations, avoiding common errors or the applied naming convention. Please note the following important classes (subclasses of the standard ontology class `owl:Thing`) or groups of classes defined:

- `AuxiliaryConcept` class, representing usually enumerative subclasses whose individuals are mainly used for knowledge organization and retrieving;
- `CCSecComponent` class, expressing security requirements: assurance requirements – SAR (`SARComponent`) and functional requirements – SFR (`SFRComponent`) defined in [1] and discussed in [23-26];
- `EnhancedGeneric` class, representing enhanced generics used as specification items for development stages other than the security requirements elaboration, defined previously for the ITSDF framework [20-21] and discussed in [23-26];
- group of classes concerning evidences discussed here; `EvidenceDoc`, represents the TOE evidences as a whole, integrating their family evidences elaborated for particular assurance families (expressed by the `FamilyEvidence` class) on the patterns basis (expressed by the `EvidenceTemplate` subclasses) and with the use of guidance documents (expressed by the `EvidenceGuide` subclasses); these issues are discussed in this paper;
- group of classes concerning particular kinds of security specifications [1] (`SecurityTarget`, `ProtectionProfile`, `LowAssST`, `LowAssPP`) and

their parts (ST_PP_Part), e.g.: SecProblemDef, the SecObjectives, SecRequirem, TSS_TOESumSpec, defined in [22] and currently integrated with SMO.

The paper is focused on the EvidenceDoc class and related ones. For this reason other ontology items will be discussed very briefly to create a common picture of the SMO issues only.

The AuxiliaryConcept subclasses are varied. For example, one of its subclasses is EAL. It contains EALs definitions as its individuals (EAL1, EAL1plus, EAL2, … EAL6plus, EAL7). The Projects subclass of AuxiliaryConcept has instances related to the particular TOE projects performed with the use of the SMO ontology, e.g. here discussed *MyFirewall* project.

The specification means representing the IT security development subdomain encompassing the CC-defined functional and assurance components and introduced enhanced generics are discussed in [23-26]. Because the evidences are implied by SARs, their short presentation is necessary. SARComponent encompasses all CC assurance classes ([1]/Part 3): ADVClass, AGDClass, ALCClass, etc. Each CC assurance class has its CC assurance families, e.g. ADVClass has ADV_ARC, ADV_FSP, ADV_IMP, etc. Similarly, the hierarchy of SFRs is expressed.

The paper is focused on the ontological representation of evidences which are elaborated for the CC-compliant evaluation process. The relations between EALs and their components are shown in Table 1 [1]. Please note that the assurance components of the given assurance family are ordered hierarchically [1] – component numbers, e.g.: ADV_FSP.1, ADV_FSP.2, …, ADV_FSP.6 are growing from the left to the right. The rigour and depth of evaluation, related to them, are increasing and accumulating while going from EAL1 to EAL7.

For the given EAL, especially for those of a lower range, some assurance families are not represented by their components (grey-marked table cells). A bold faced number in the mentioned table means that a component is introduced or a component of the lower rigour is replaced by a component of the higher rigour. All these rules specified in the standard are expressed by the elaborated ontology.

Fig. 1 presents the organization of evidence-related items in the Protégé tool. The proposed idea is based on three hierarchy levels:

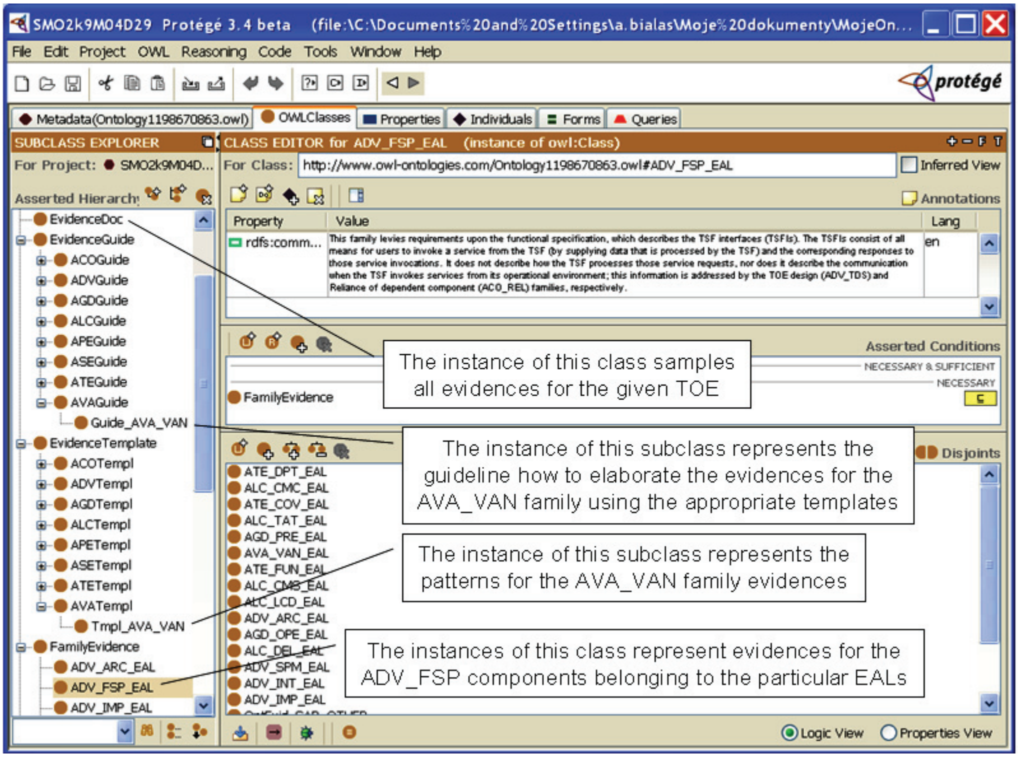
- TOE evidences as a whole, composed properly from families evidences (depending the EAL, required SARs addition and/or substitution),
- particular families evidences depending on EAL (please note Table 1),
- templates and guidelines used to elaborate families evidences.

Assurance class	**Assurance Family**	**Assurance Components by Evaluation Assurance Level**						
		EAL1	**EAL2**	**EAL3**	**EAL4**	**EAL5**	**EAL6**	**EAL7**
ADV	ADV_ARC		**1**	1	1	1	1	1
	ADV_FSP	**1**	**2**	**3**	**4**	**5**	5	**6**
	ADV_IMP				**1**	1	**2**	2
	ADV_INT					**2**	**3**	3
	ADV_SPM						**1**	1
	ADV_TDS		**1**	**2**	**3**	**4**	**5**	**6**
AGD	AGD_OPE	**1**	1	1	1	1	1	1
	AGD_PRE	**1**	1	1	1	1	1	1
ALC	ALC_CMC	**1**	**2**	**3**	**4**	4	**5**	5
	ALC_CMS	**1**	**2**	**3**	**4**	**5**	5	5
	ALC_DEL		**1**	1	1	1	1	1
	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR	optional at any EAL						
	ALC_LCD			**1**	1	1	1	**2**
	ALC_TAT				**1**	**2**	**3**	3
ATE	ATE_COV		1	**2**	2	2	**3**	3
	ATE_DPT			**1**	**2**	**3**	3	**4**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
AVA	AVA_VAN	**1**	**2**	2	**3**	**4**	**5**	5

Tab. 1. Evaluation assurance level summary – TOE related components [1] (Part 3)

The classes dealing with evidences are shown in the "Protégé Subclass Explorer" window (the left part of Fig. 1). The evidence documentation for an IT product or system with respect to the declared EAL is represented by the instance of the `EvidenceDoc` class. This instance integrates evidences implied by particular assurance families, which are expressed by the `FamilyEvidence` subclasses (exactly: by their instances): `ADV_ARC_EAL`, `ADV_FSP_EAL`, `ADV_IMP_EAL`, …, `AVA_VAN_EAL`, `OptEvid_ALC_FLR`, `OptEvid_SAR_OTHER` classes. The last two subclasses of `FamilyEvidence` have special meaning. The first one, the `OptEvid_ALC_FLR` class, expresses flaw remediation requirements that can be included optionally for any EAL, while the second one, i.e. the `OptEvid_SAR_OTHER` class, represents evidences added by developers for the user's defined SARs.

The family evidences are elaborated on the patterns basis, expressed by the `EvidenceTemplate` subclasses, while the `EvidenceGuide` subclasses express guidelines how to use these patterns.



This family levies requirements upon the functional specification, which describes the TSF interfaces (TSFIs). The TSFIs consist of all means for users to invoke a service from the TSF (by supplying data that is processed by the TSF) and the corresponding responses to those service invocations. It does not describe how the TSF processes those service requests, nor does it describe the communication when the TSF invokes services from its operational environment; this information is addressed by the TOE design (ADV_TDS) and Reliance of dependent component (ACO_REL) families, respectively.

The instance of this class samples all evidences for the given TOE

The instance of this subclass represents the guideline how to elaborate the evidences for the AVA_VAN family using the appropriate templates

The instance of this subclass represents the patterns for the AVA_VAN family evidences

The instances of this class represent evidences for the ADV_FSP components belonging to the particular EALs

Fig. 1. Examples of SMO ontology classes representing evidences within the Protégé [18] environment.

In the right part of Fig. 1 ("Class Editor" window) some details are shown about the highlighted ADV_FSP_EAL class, dealing with the TOE functional specification. The upper part includes the related annotation-type property `rdfs:comment`, the middle one – the Protégé domain, range and restriction facilities (see the next subsection), and the bottom part – the defined disjoints of the highlighted class.

Classes on the same generality level, usually disjoined, are called siblings, e.g. particular kinds of patterns of evidences. For example, the evidences for the security policy modelling (`ADV_SPM_EAL`) can express neither the TOE architecture (`ADV_ARC_EAL`) nor guidance documentation (`AGD_OPE_EAL`). Please note that disjoined classes cannot have common individuals. The classes representing evidences for particular assurance families are disjoined too.

4.5. The class properties and their restrictions

The hierarchy of classes defines the general taxonomy of the ontology concepts. The next step is to define the class properties. There are some general principles with respect to properties. Because all subclasses of a given class inherit the properties of that class, a property representing the given class property should be placed on the highest possible level of the class hierarchy. Besides, when a class has multiple superclasses it simply inherits properties from all of them. The classes to which an instance-type property is attached are called a class domain, while the classes indicated by this property are called a class range. The possible values of the property can be refined by defining the restrictions for them. The restrictions describe or limit the set of possible values for the given property.

Three kinds of standard properties [3], [18] are used:

- object (called also "instance-type") properties, expressing "complex properties", i.e. relationships between an individual member (instance) of the given class (the object) and other instances; e.g. when the given instance consists of other instances or points to other instances; examples: the `assignedToProject` property specifies a project name (`Project` class range) to which given ontology item belongs (in this case domain encompasses almost all ontology classes), the `hasBasicEvidence` property assigns assurance family evidences (`FamilyEvidence` range) to the composed set of evidences for the TOE (`Evidences` domain);

- data-type properties, expressing "simple properties" or "attributes", i.e. intrinsic or extrinsic properties of the instances of the most elementary classes; the data type used for this property can be any of those commonly used in modelling or programming, e.g.: integer, byte, float, time, date, enumeration, string; examples: the `hasComments` property, representing verbal notes (the range string) added to instances of some classes related to evidences (in this case domain is a sum: `EvidenceDoc or FamilyEvidence or EvidenceGuide`), the properties: `hasTitle`, `fileName`, `fileLocation` (domain: `EvidenceDoc or FamilyEvidence or EvidenceTemplate or EvidenceGuide` and the range `string`) are used to reference external documents;

- annotation properties expressing the meaning of the given class; they are RDF-based and are used to document different ontology items (classes, properties, instances); example: the `rdfs:comment` property gives more explanation of the given ontology item (Fig. 1).

The presented there SMO ontology uses all kinds of the above properties.

4.6. Creating instances and filling in their properties

As it was mentioned earlier, when the ontology classes are defined, an important issue is to identify classes which can have instances. In the SMO ontology they belong to the lowest levels of the class hierarchy and encompass: functional components, assurance components, enhanced generics and the discussed here evidences. The SMO development has enabled the elaboration of a knowledge base encompassing all CC-defined functional components (about 132 items), assurance components (about 86 items) and the author's defined enhanced generics (about 350 items) designed to specify the security problem definition, security objectives for the TOE and its environment and security functions. Together they contain items needed to specify security targets (or protection profiles) for many different IT products or systems (TOE). The knowledge database contains also discussed here evidence templates, currently elaborated for the most frequently used assurance components from the middle range of the EAL scale.

The knowledge database can be used as a library of predefined specification means and evidence items allowing to retrieve solutions for elementary security issues during the IT security- and TOE development processes. The ontology definition may be considered complete, including a basic set of instances. However, some properties, especially data- and annotation properties, usually remain empty due to the extensive character of the work and the encountered bottleneck, i.e. manually performed knowledge acquisition.

Some general rules of the ontology engineering should be taken into consideration for the instances too. They differ a little from the rules of the object modelling domain. Please note that an instance of a subclass is an instance of a superclass.

The instances representing assurance family evidences need extra explanation. They are created according to the earlier mentioned Table 1. For each of the classes, representing assurance family evidences shown in Fig. 1 (`ADV_ARC_EAL`, `ADV_FSP_EAL`, `ADV_IMP_EAL`, …, `AVA_VAN_EAL`), appropriate instances are created, i.e. in the situation when an assurance component appears or is replaced by a more restrictive one. For example, the `ADV_ARC_EAL` class has only one instance: `ADV_ARC_EAL_2`. Please note that for EAL1 no ADV_ARC components are used. The first of them appears for EAL2 and is used until EAL7. The `ADV_FSP_EAL` class has the following instances: `ADV_FSP_EAL_1` (for EAL1), `ADV_FSP_EAL_2` (for EAL2), `ADV_FSP_EAL_3` (for EAL3), `ADV_FSP_EAL_4` (for EAL4), `ADV_FSP_EAL_5` (for EAL5 and EAL6) and `ADV_FSP_EAL_7` (for EAL7). The `ADV_SPM_EAL` class has only one instance (`ADV_SPM_EAL_6`), covering EAL6 and EAL7. Please note that the index included in the name of an instance expresses the starting EAL and it is valid until EAL7, unless a more rigorous component appears meanwhile. `OptEvid_ALC_FLR` and `OptEvid_SAR_OTHER` are considered individually.

4.7. Testing and validation of the developed ontology

The created ontology should be tested during its development and finally validated by the users. The Protégé environment provides some facilities to perform these operations. The ontologies are vulnerable to some commonly known errors [3], such as: cycles in the class hierarchy, violation of property constraints, interval restrictions issuing empty intervals, e.g. min val. > max val., terms not properly defined, classes with a single subclass, classes and properties with no definitions, properties with no constraints like value type or cardinality. They can be detected with the use of the Protégé menu functions, like "Checking consistency", "Run ontology tests" or by manual ontology inspections. Besides, the usability tests can be performed, for example: checking if the right structures of instances are composed, if they have assumed properties, if needed information can be retrieved properly by queries from the knowledge database. More advanced usability tests are carried out during the ontology validation.

On one hand, ontology designing is subjective and many different correct solutions are possible. On the other hand, the ontology ought to be objectively correct [3]. The ontology validation on the near realistic *MyFirewall* project, based on the example described in [21], [27], helps to decide which solutions are acceptable and which need corrections and further development with respect to the ontology users' needs and expectations. This validation encompasses two issues: building the ST with the use of properly selected specification means [23-24] and the presented composing of evidences for the worked-out ST. During the SMO validation different queries will be issued and the usability of their results will be assessed. The validation process is rather extensive and will be presented in a separate publication.

5. Summary

The paper presents multidisciplinary research and development works encompassing mainly the security engineering and knowledge engineering domains. It concerns the ontological approach to the IT security development and implementation processes compliant with the Common Criteria standard. The Specification Means Ontology encompasses the following:

- items discussed in [23-24], used for the security targets (or protection profiles) specification during IT security development, i.e.: all functional and assurance security components defined by Common Criteria v. 3.1 and author's defined, enhanced generics for threats, OSPs, assumptions, security objectives, and functions,

- items discussed in the paper, concerning evidences elaborated on the ST basis during the IT product or system (TOE) development and provided for the IT security evaluation process.

Based on the general background of SMO and the related knowledge base features, the paper focuses on the issues concerning evidences, which are elaborated for the given IT product or system according to the EAL claimed for it, and later, are independently evaluated together with the TOE. The paper gives a proposal how to organize the evidences and the evidences elaboration process using an ontological approach.

Summing up, the major contribution of the paper is to provide an ontology-based method and tool to elaborate and manage evidences which are developed for the Common Criteria evaluation process for different kinds of IT products or systems and the claimed EAL.

Generally, TOE evidences are currently elaborated by trained developers co-operating with external consultants providing them with the proper know-how [2], however, the balance between the involvement of the developers and consultants may differ, depending on earlier performed projects, gained experiences, organization policy with respect to intellectual property rights retention and to the CC-related competency for future projects retention. Apart from the Common Criteria standard [1], the general guide on ST/PP [27] and the guide on the evidences [28], developers are not provided with more enhanced knowledge, well structured evidence patterns, specialized supporting tools, clear procedures (methods) helping them in the step-by-step elaboration of evidences. They use, as supporting tools, text editors, partially CAD/CAE systems, because there are only few specialized CC tools available. CC-related works (IT security- and TOE development, IT security evaluation) are rather poorly automated in comparison with other engineering domains. The developers try to elaborate their own methods or simply use expensive know-how [2]. These factors make the work difficult for the developers, raise development costs and create a barrier to broader deployment of dependable IT solutions. The shortage of security expertise of IT developers, especially those who elaborate software products, may cause security problems. Security related knowledge provided for these people may be considered another source of product assurance.

By applying a knowledge engineering approach to the Common Criteria domain, the author's works aim at providing developers with: design patterns, methodology, tools and related knowledge, which all help to elaborate evidences.

Only few research works try to apply knowledge engineering methodology to manage Common Criteria related knowledge. Among the works identified in Section 2, it is worth mentioning an extended knowledge base designed for Common Criteria developers from Soka University, Japan [29], though it does not represent an ontological approach.

The development of the SMO ontology is a very extensive task covering different issues. The main directions of the planned works are: refining the internal knowledge models to allow more sophisticated competency questions, improving knowledge acquisition and multi-project management facilities, integration of selected external ontologies, and integration with advanced knowledge management systems. This iterative and incremental process needs permanent tests, validations and knowledge base optimization on real projects.

SMO should be validated with the use of different projects. One of the planned case studies was completed and its result were published in [25-26]. It concerns a motion sensor of a digital tachograph compliant with the EC regulations [30], [31]. Two another validations dealing with a medical diagnostic IT product [32] and an intelligent sensor for gas monitoring application [19] have been completed. These both works were summarized in [33] presenting a coherent set of security target related design patterns for intelligent sensors. All above mentioned extensive validations are focused on the IT security development process. The SMO validation with respect to the TOE development process will be discussed in a separate publication.

The results of these researches are used in the CCMODE R&D Project (Common Criteria compliant, Modular, Open IT security Development Environment) carried out by the Institute of Innovative Technologies EMAG [34]. The objective of the project is to work out a methodology and tools to develop and manage development environments of IT security-enhanced products and systems for the purposes of their future certification.

Acknowledgments

This work was conducted using the Protégé resource, which is supported by the grant LM007885 from the United States National Library of Medicine.

The results of the paper were used in the project CCMODE co-financed from EU Resources within European Regional Development Fund (UDA POIG 01.03.01.156/08).

References

1. *Common Criteria for IT security evaluation*, part 1-3. v. 3.1. (2009)
2. W. H. Higaki, *Successful Common Criteria Evaluation. A Practical Guide for Vendors*. Copyright 2010 by Wesley Hisao Higaki, Lexington, KY 2011.

3. N. F. Noy, D. L. McGuiness, *Ontology Development 101: A Guide to Creating Your First Ontology*. In: Stanford Knowledge Syst. Lab. Tech. Rep. KSL-01-05 and Stanford Medical Informatics Tech. Rep. SMI-2001-0880. Stanford University, CA. http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html (2001). Accessed March 2013

4. D. S. Yavagal, S. W. Lee, G. J. Ahn, R. A. Gandhi, *Common Criteria Requirements Modeling and its Uses for Quality of Information Assurance (QoIA)*. In: Proc. of the 43rd Annual ACM Southeast Conference (ACMSE‘05), Vol. 2, ISBN:1-59593-059-0, pp. 130-135. Kennesaw State University Kennesaw, Georgia, ACM New York (2005)

5. A. Ekelhart, S. Fenz, G. Goluch, E. Weippl, *Ontological Mapping of Common Criteria's Security Assurance Requirements*. In: H. Venter, M. Eloff, L. Labuschagne, J. Eloff, von R. Solms (eds.) New Approaches for Security, Privacy and Trust in Complex Environments, pp. 85-95. ISBN 978-0-387-72366-2, Springer, Boston (2007)

6. *Secure Business. Common Criteria ontology*. http://research.securityresearch.at/research/focus/common-criteria-security-assurance/ (2008). Accessed October 2008

7. A. Vorobiev, N. Bekmamedova, *An Ontological Approach Applied to Information Security and Trust*. In: Proc. of the 18th Australasian Conf. on Information Systems, Toowoomba. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.8433&rep=rep1&type=pdf (2007). Accessed March 2013

8. A. Kim, J. Luo, M. Kang, *Security Ontology for Annotating Resources, On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, In: Proceedings part II, pp. 1483-1499, Agia Napa, Cyprus, October–November 2005, Lecture Notes in Computer Science (LNCS), ISBN 978-3-540-29738-3, Springer, Berlin; Heidelberg (2005).

9. A. Ekelhart, S. Fenz, G. Goluch, B. Riedel, et al., *Information Security Fortification by Ontological Mapping of the ISO/IEC 27001 Standard*. In: Proc. of the 13th Pacific Rim Int. Symposium on Dependable Computing, Washington DC, USA, pp. 381-388. IEEE Computer Society. http://publik.tuwien.ac.at/files/pub-inf_4689.pdf (2007). Accessed March 2013

10. L. A. F. Martimiano, E. S. Moreira, *Using ontologies to assist security management*. In: Proc. of the 8th Intl. Protégé Conference, Madrid. http://www.ppgia.pucpr.br/~maziero/pesquisa/ceseg/sbseg06/conteudo/artigos/resumos/19513.pdf (2005). Accessed June 2008, moved to: http://www.redes.unb.br/ceseg/anais/2006/conteudo/artigos/resumos/19513.pdf Accessed March 2013

11. A. Ekelhart, S. Fenz, M. Klemen, E. Weippl, *Security Ontologies: Improving Quantitative Risk Analysis*. In: Proceedings of the 40th Hawaii International Conference on System Sciences, Big Island, Hawaii, ISBN: 0-7695-2755-8, IEEE Computer Society Press. (2007)

12. B. Tsoumas, S. Dritsas, D. Gritzalis, *An Ontology-Based Approach to Information Systems Security Management*. In: Proc. of 3rd International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2005, ISBN 978-3-540-29113-8, St. Petersburg, Russia, September 2005, Lecture Notes in Computer Science (LNCS), Volume 3685/2005, pp. 151-164. Springer, Berlin; Heidelberg (2005)

13. L. A. F. Martimiano, E.S. Moreira, *An OWL-based Security Incident Ontology*. In: Proc of the 8th Intl. Protégé Conf. Madrid. http://protege.stanford.edu/conference/2005/submissions/posters/poster-martimiano.pdf (2005). Accessed Feb 2013

14. *CSL – Computer Science Laboratory. Security ontologies in OWL*. http://www.csl.sri.com/users/denker/owl-sec/ontologies/ (2010). Accessed June 2008, moved to: http://www.csl.sri.com/people/denker/ Accessed March 2013

15. *DAML Services – Security and privacy*. http://www.daml.org/services/owl-s/security.html (2001-2013). Accessed March 2013

16. *Herzog's Security Ontology*. Linköping University. http://www.ida.liu.se/~iislab/projects/secont/ (2007). Accessed Jan 2013

17. *REI Ontology Specifications. ver. 2.0*. University of Maryland. http://www.csee.umbc.edu/~lkagal1/rei/ (2010). Accessed Jan 2013

18. *Protégé Ontology Editor and Knowledge Acquisition System, v.3.4*. Stanford University. http://protege.stanford.edu/ (2008). Accessed May 2008, March 2013

19. A. Bialas, *Common Criteria Related Security Design Patterns—Validation on the Intelligent Sensor Example Designed for Mine Environment*. Sensors 2010, 10, 4456-4496. available at: http://www.mdpi.com/1424-8220/10/5/4456 Accessed Jan 2013

20. A. Bialas, *Semiformal Approach to the IT Security Development*. In: W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak (Eds.) Proc. of the Int. Conf. on Dependability of Computer Systems DepCoS-RELCOMEX 2007, pp. 3-11. ISBN 0-7695-2850-3, IEEE Computer Society, Los Alamitos; Washington; Tokyo (2007)

21. A. Bialas, *Semiformal Common Criteria Compliant IT Security Development Framework*. Studia Informatica vol. 29, Number 2B(77), Silesian University of Technology Press Gliwice. http://www.znsi.aei.polsl.pl/ (2008). Accessed March 2013

22. A. Bialas, *Ontology-based Approach to the Common Criteria Compliant IT Security Development*. In: H. Arabnia, S. Aissi, M. Bedworth (eds.) Proc. of the 2008 Int. Conf. on Security and Management, pp. 586-592. CSREA Press, Las Vegas (2008)

23. A. Bialas, *Ontology-based Security Problem Definition and Solution for the Common Criteria Compliant Development Process*. In: Proceedings of 2009 Fourth International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX 2009), pp. 3-10. ISBN 978-0-7695-3674-3, IEEE Computer Society, Los Alamitos; Washington; Tokyo (2009)

24. A. Bialas, *Validation of the Specification Means Ontology on the Simple Firewall Case*. In: H. Arabnia, K. Daimi (Eds.), Proc. of the 2009 Int. Conf. on Security and Management, (WORLDCOMP'09 – The 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing), Vol. I, pp. 278-284. ISBN: 1-60132-124-4, 1-60132-125-2 (1-60132-126-0, CSREA Press, Las Vegas (2009)

25. A. Białas, *Security-related design patterns for intelligent sensors requiring measurable assurance*. Electrical Review (Przegląd Elektrotechniczny), ISSN 0033-2097, vol. 85 (R.85), Number 7/2009, 92-99. Sigma-NOT, Warsaw (2009)

26. A. Białas, *Ontological approach to the motion sensor security developmen*t. Electrical Review (Przegląd Elektrotechniczny), ISSN 0033-2097, vol. 85 (R.85), Number 11/2009, 36-44. Sigma-NOT, Warsaw (2009)

27. ISO/IEC TR 15446. Guide for the production of protection profiles and security targets (2009)

28. BSI Guidelines for Developer Documentation according to Common Criteria Version 3.1. Bundesamt für Sicherheit in der Informationstechnik, Bonn (2007)

29. G. H. Ramirez-Caceres, Y. Teshigawara, *Design and Development of a Knowledge-based Tool Based on Multiples International Standard*s, The 8th Int. Common Criteria Conference, Rome, 25-27 September 2007. http://www.8iccc.com/index.php (2007). Accessed April 2009

30 Commission Regulation (EC) No.1360/2002 on recording equipment in road transport, Annex 1B Requirements for Construction, Testing, Installation and Inspection. Official Journal of the European Communities, L 207, 204-252, Commission of the European Communities (2002)

31. I. Furgel, K. Lemke, *A Review of the Digital Tachograph System*. In: Embedded Security in Cars, ISBN: 978-3-540-28384-3, pp. 69-94. Springer, Berlin Heidelberg (2006) 32. Bialas, A. Intelligent Sensors Security. Sensors 2010, 10, 822-859. available at: http://www.mdpi.com/1424-8220/10/1/822/ Accessed Jan 2013

32. A. Bialas, *Common Criteria Related Security Design Patterns for Intelligent Sensors—Knowledge Engineering-Based Implementation*. Sensors 2011, 11, 8085-8114. available at: http://www.mdpi.com/1424-8220/11/8/8085/ Accessed Jan 2013

33. CCMODE (Common Criteria compliant, Modular, Open IT security Development Environment). Project co-financed from EU Resources within European Regional Development Fund (UDA POIG 01.03.01.156/08; http://www.commoncriteria.pl/. Accessed 20 March 2013

Model materiału dowodowego do oceny zabezpieczeń według metodyki Wspólne Kryteria bazujący na ontologii

Streszczenie

Artykuł przedstawia wybrane zagadnienia dotyczące modelu materiału dowodowego wykorzystywanego w procesie oceny i certyfikacji zabezpieczeń informatycznych. Model opracowano w oparciu o metody inżynierii wiedzy i metodykę "Wspólne Kryteria" (ISO/IEC 15408 Common Criteria). Zakres i szczegółowość materiału dowodowego, przedkładanego wraz z produktem informatycznym (sprzęt, oprogramowanie, w tym układowe, system informatyczny) do oceny w niezależnym, akredytowanym laboratorium, są implikowane przez zadeklarowany dla produktu poziom uzasadnionego zaufania EAL (*Evaluation Assurance Level*). EAL1 to wartość minimalna, EAL7 – maksymalna. Każdemu z poziomów EAL odpowiada pewien spójny zbiór wymagań uzasadniających zaufanie, czyli pakiet komponentów SAR (*Security Assurance Requirement*) – Tab.1. Większość z ponad tysiąca ocenionych produktów posiada certyfikaty EAL3-EAL4. Oceniany pod względem wiarygodności swych zabezpieczeń, produkt informatyczny zwany jest przedmiotem oceny (*TOE – Target of Evaluation*). Na wstępie należy dla niego opracować materiał dowodowy o nazwie zadanie zabezpieczeń (*ST – Security Target*), który stanowi podsumowanie przeprowadzonych analiz bezpieczeństwa produktu i zawiera wykaz funkcji zabezpieczających, które należy zaimplementować w produkcie informatycznym na zadeklarowanym arbitralnie dla niego poziomie EAL, by zasoby informacji były w wystarczający sposób chronione przed zagrożeniami. W drugim etapie wypracowywany jest obszerny materiał dla samego produktu (projekt TOE, jego interfejsów, sposób implementacji, dokumentacja uruchomieniowa i użytkowa, testy, ocena podatności, itp.) i środowiska rozwojowego, w którym ten produkt powstaje (procesy rozwojowe w cyklu życia, zabezpieczenia środowiska rozwojowego, narzędzia, zarządzanie konfiguracją, usterkami i dostawą dla użytkownika, itp.). Artykuł zawiera wprowadzenie do metodyki Common Criteria, przegląd dotychczasowych badań, w tym badań własnych, w zakresie ontologii, modelowania pojęć i procesów tej metodyki.

Całość metodyki Common Criteria zawarto w modelu wyrażonym za pomocą ontologii środków specyfikacji (*SMO – Specification Means Ontology*), jednak w głównej części artykułu (Rozdział 4) uwagę skupiono na fragmencie modelu odnoszącym się do materiału dowodowego dla samego TOE (bez ST). Przedstawiono proces tworzenia ontologii zgodnie z klasycznym podejściem [3] i z wykorzystaniem popularnego narzędzia [18]. Pokazano, jak opracowano rozbudowaną hierarchię klas, opisano własności klas i ich ograniczenia, a także, jak tworzono bazę wiedzy zawierającą środki specyfikacji. Artykuł rozwiązuje problem organizacji (struktury i zawartości) wzorców materiału dowodowego, tworząc dla nich szablony i instrukcje wypełnienia

ich treścią dotyczącą opracowywanego produktu informatycznego. Opracowaną ontologię poddawano testom w toku tworzenia.

Ontologię SMO wykorzystano więc jako model materiału dowodowego, implikowanego przez komponenty SAR należące dla poszczególnych pakietów EAL. Model ten, po poddaniu go walidacji i rewizji, może być podstawą do budowy aplikacji użytkowych dla twórców materiału dowodowego.