Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 1135–1140

# Software Implementation of Common Criteria Related Design Patterns

Dariusz Rogowski
Institute of Innovative Technologies EMAG ul. Leopolda 31, 40-189 Katowice, Poland
Email: drogowski@emag.pl

***Abstract*—Writing evidence documents for evaluation and certification processes according to the Common Criteria security standard is a very difficult, time-consuming and complex task. Nowadays there are only a few, limited solutions based on templates and software tools which can efficiently support developers in preparing evaluation deliverables. This paper describes the results of an R&D project whose aim was to work out a computer-aided tool with built-in design patterns. Firstly, according to all security assurance requirements the design patterns in a paper version were prepared. Secondly, they were verified and validated by the developers in order to make some amendments and improvements. The conclusions were used as the source of functional requirements for a computer-aided tool. As a result a complete computer system was designed which implements the design patterns, knowledge base, evaluation methodology, and additional external supporting software. That solution facilitates and speeds up the development of the evidence documentation.**

## I. Introduction

SECURITY features of IT products have received much attention in recent years due to quickly rising numbers of cyber-attacks on important data and information. That is why there is a big demand coming from governments and private users for trusted IT products countering such threats. These products can be more reliable thanks to the evaluation and certification of their built-in security functions. Assessment processes should be conducted by an independent licensed laboratory which can use a security standard with requirements for a product development and documentation.

Here, the Common Criteria for Information Technology Security Evaluation standard (referred to as "Common Criteria" or "CC" throughout this paper), also known as ISO/IEC 15408 [1]–[3], provides a set of development rules and evaluation requirements [4] for the security measures applied in IT products. The results of CC-based evaluation are accepted in countries which joined the Common Criteria Recognition Arrangement (CCRA). This arrangement allows end users to recognize certificates regardless of the country in which they were issued. Therefore the certified products of different vendors can be easily compared by the users which can choose the best option. On the other hand we should remember that CC does not define the product security features or functionality but it provides assurance that the process of specification, implementation and evaluation of the product has been made in a rigorous manner. This assurance is assigned to one of seven assurance levels reflecting the requirements met in the development process of the product.

The product is the subject of the evaluation against the given EAL requirements and then it is called the Target of Evaluation (TOE). Evaluation Assurance Level (EAL) reflects the degree of confidence a user can have in the results of the evaluation and performance of the TOE. The lower assurance levels, EAL 1 through 4, concern most products, and do not require evaluation of the software, only of the development process and documentation. These lower levels are recognized under CCRA whereas the higher EALs are generally country-specific [5] and require a source code of the product to be analyzed.

However, it has been found by many developers of IT secure products that preparation for the Common Criteria evaluation process is very difficult, time-consuming and needs a lot of knowledge of CC requirements, all due to the fact that a lot of evidence documents have to be prepared. That problem is very important and has to be solved in order to make the whole evaluation process cost-effective and developers-friendly.

One solution was based on a series of guidelines and supporting documents issued by German Federal Office of Information Security (BSI) – the leader of researches in the field of the Common Criteria standard. This guidance documentation gives some advice on the structure and contents of evidence documents. Some templates were issued and could be used by developers but still too much work has to be done on their own [6], [7].

The second solution was based on very few software applications which could support users in the preparation of evidence documents and security development process. Unfortunately, these applications provided only basic functionality. They are limited to making only two main security specification documents (Security Target – ST, Protection Profile – PP), discussed later in this paper [8], [9]. Moreover, some of the software tools are not supported and developed by their producers any more.

Although the solutions mentioned above were offered a few years ago, relatively little attention has been paid to other evidence documents needed for the evaluation process. The guides and computer-aided tools are focused mainly on the preparation of STs and PPs documents. Apart from that, there is weak integration of the guidance knowledge within the software tools. In addition, if the developers want to create the evidence document only by using the guidelines and

978-1-4673-4471-5/$25.00 © 2013, IEEE

templates, they still have a lot of work to do by themselves. They have to plan the structure of the document and find out what kind of information they should write down in the given section. That is why preparing the documentation is still difficult and not effective enough to encourage the developers to do this task.

This paper presents a complete and integrated solution which was worked out in the CCMODE R&D project (Common Criteria compliant, Modular, Open IT security Development Environment) carried out by the Institute of Innovative Technologies EMAG. The aim of the project was to work out a methodology and tools to develop and manage development environments of IT security-enhanced products for the purposes of their future Common Criteria certification. As a result a set of design patterns (the core of the methodology) was developed and next implemented in the computer-aided system CCMODE Tools. Thanks to the software implementation of the design patterns the developers receive one complete solution which facilitates production processes of the TOE and related documentation.

The paper is organized as follows. Section II presents the state of the art. Section III describes shortly the basic evidence documents required for the CC evaluation process. Section IV explains the methodology used for working out the design patterns and their implementation into the computer tool. Section V gives an overview of the CCMODE Tools main modules and their functionality used for evidence documents preparation. Section VI contains conclusions and experiences gained during the usage of the tool with built-in design patterns.

## II. State of the art

The best starting point for building the design patterns is the Common Criteria standard which comprises three parts [1]–[3]. The current version of CC was issued in 2012. The first part is a general introduction to the CC methodology with explanation of basic terms and definitions. The second part describes security functional requirements which determine the desired security behavior of a TOE. The third part, the most important for building the patterns, defines the assurance requirements for a TOE and evaluation criteria for PPs, STs and other evidence documents. There are many companion documents to the CC standard. One of them is the Common Evaluation Methodology (CEM) [4] which helps evaluators to conduct the TOE assessment process. It defines evaluation activities to be done by the evaluators and presents work units – the most granular level of evaluation work – that help to issue verdicts about the quality of security implemented in the TOE. Other documents like technical reports and users guides explain step by step how to build evidence documentation. For instance, the ISO/IEC Technical Committee for Information Technology issued a technical report that is a guide for the production of PPs and STs [10]. This report provides methodologies, techniques and practical tips that developers can use to prepare security specification documents in an efficient and consistent manner. BSI issued a guide for developers of the STs and PPs [11]. Apart from that there is a guide that offers assistance to less experienced developers by extracting the information about the evidence from CC [12]. It explains requirements concerning the structure and contents of documents to be provided for the CC evaluation process. Another guide concerns evaluation reports according to CC and gives some advice and recommendations on the structure of information provided in these reports [13]. The CC standard and all guides mentioned above are used by the developers in common practice for writing evidence documents. But this way of work is very inconvenient because the developers must carefully read recommendations, check requirements and think about the necessary information to be provided in every new document each time they begin a project.

Although this guidelines-based approach helps the developers to work out documentation, it still does not allow to get rid of inefficient and time-consuming work. That is why some software aiding tools were applied to enhance the work with design patterns. Most of the software tools are dedicated only to preparing security specification documents (ST, PP) [14]. For example, an MS Windows application "CC Toolbox" sponsored by the National Information Assurance Partnership (NIAP, the US government initiative) used to assist users in writing ST and PP but is not longer supported and available. In [15] a generator of security target templates, named "GEST" was presented that can automatically generate security target templates from already evaluated and certified security targets. One of the Spanish CC licensed laboratories "Applus" presented a tool that reduces and automates some developer's activities of evidence documents preparation [8]. Another software tool, "TL SET", was introduced by Trusted Labs [16]. It is a smart editor for Security Targets and Protection Profiles. It integrates predefined libraries of the Common Criteria functional and assurance requirements and a user-friendly graphical interface to fill out the documents. There are also tools with built-in OWL language (OWL – Web Ontology Language) [17]–[20]. These tools are dedicated to build functional specification of the TOE and security problem definition.

So far all the solutions based on guidelines and computer tools have concerned mainly two basic documents: ST and PP. This paper presents a solution which allows to produce all the necessary documents and uses the context-sensitive help based on the CC standard and supplementary documents. The next section describes how complex a task of preparing all the evidence documents can be due to the fact that several documents have to be created.

## III. Evidence documents

In the Common Criteria evaluation process security functions of the TOE are evaluated according to security assurance requirements (SARs) in the given EAL. Many documents should be prepared for the needs of the evaluation.

The most important is the Security Target (ST). The ST describes a specific TOE and is written by the developer. The ST can be based on a document called the Protection Profile (PP). The PP describes the general requirements for a TOE type and is used as a template for many different ST

documents. The ST consists of a security problem definition; security objectives; security requirements; a summary specification – showing how the security functions are implemented in the TOE. The ST document claims conformance with the declared EAL and this determines all requirements which have to be fulfilled by the product and described in evidence documentation.

The EAL package consists of assurance components which are organized into classes and families. The following descriptions of classes also include their short names (in brackets) which are commonly used in the CC standard. The Protection Profile Evaluation (APE) and Security Target Evaluation (ASE) classes describe the content and presentation of the PP and ST documents. The Development (ADV) class encompasses six families and nineteen components; it provides information about structuring of the TOE security functionality. The Guidance Documents (AGD) class is divided into two families (with one component for each family); it provides the requirements for preparative and operational user guides. The Life-cycle Support (ALC) class consists of seven families and twenty one components; it concerns the aspects of establishing discipline and control in the TOE development and maintenance during its whole life-cycle. The Tests (ATE) class encompasses four families and twelve components; it provides assurance that the TOE security functions were tested and they operate according to their design descriptions. The Vulnerability Assessment (AVA) class has only one family with five components; it addresses the possibility of exploitable vulnerabilities introduced in the TOE or in its development or operational environment. The Composition (ACO) class encompasses five families and eleven components; it assures that the TOE composed of other evaluated TOEs will operate securely. For instance, the EAL 3 package has fifteen components and for each one a proper evidence document has to be prepared.

On the basis of all assurance components taken from EAL packages the design patterns were worked out and then implemented into the computer tool.

## IV. Methodology

In the first part of the CCMODE project a set of design patterns in the form of MS Word documents with predefined chapters and sections was prepared. The patterns were validated and assessed by independent experts in the field. Although they assessed the patterns as very helpful, they proposed some difficult and repeatable operations which can be automated by a computer tool. These insights allowed to make some functional assumptions for the CCMODE Tools system.

In the project there were design patterns created for all components of security assurance requirements (SARs). Next the patterns were verified and validated by developers chosen from the software and hardware industry. The validation was made upon the use cases method. The developers used selected design patterns to make evidence documentation of their software and hardware IT products. As a result of the validation, necessary changes and amendments were incorporated into the patterns. Furthermore, the developers concluded that some automation features should be implemented into the patterns.

In the next project stages a prototype of the software was developed. The prototype was next validated in two selected development environments of software and hardware IT products by using the case study method. A few evidence materials were prepared by developers. Documents for the TOE (ADV class) and for the environment (ALC class) were prepared. The case studies showed what else should be implemented in the computer tool to make the work with documents more effective and easier.

As a result, the CCMODE Tools system was worked out. The system integrates: modules of the development environment management, design patterns, knowledge base, evaluation methodology, and external supporting software. The system can be integrated with other security standards, like an information security management standard (ISMS, based on ISO/IEC 27001) or business continuity management standard (BCMS, based on BS 25999) [21].

Next chapters describe functionality of the software system which implements the design patterns.

## V. Applying design patterns in a computer system

Developers use the software tool to start the project of an IT product in accordance with the chosen EAL level. They configure necessary external systems and deliver basic information about the type of the product, roles and duties of the system users, life-cycle model of the TOE, software and hardware tools used, security standards and regulations. Consequently, the following developers' actions can be automated by the software tool:

- verification of the development environment conformance with the CC standard;
- developing a security specification of the TOE in the ST document;
- providing the security problem definition that has to be solved by the TOE;
- specifying security objectives and security functional requirements to resolve the security problem;
- preparing evidence documentation with the use of the design patterns;
- defining life-cycle models for different types of IT products;
- testing the TOE and flaws remediation; establishing communication channels for flaws reports.

The actions mentioned above were next implemented in dedicated modules of the CCMODE Tools system. The following subsections describe the main modules.

### A. CCMODE Tools system

A general model of the system is depicted in Fig. 1. The model consists of the Environment Management Tool (EMT), documents generator (GenDoc), knowledge base, evaluation module, external supporting systems, optional security systems (BCMS or ISMS) which can be used as an additional source of assurance to the whole development environment [22].

EMT is the main module which supports the configuration and management of the IT products that are to be built in the development environment. EMT makes it possible to define the system users and their roles in the project, the desired EAL level and life-cycle model.

The knowledge base is a source of context-sensitive help about the CC requirements and guidelines. It includes design patterns, terms and definitions which can be obtained by other modules. It also comprises the guidelines that help to resolve typical security problems with the use of predefined security objectives, threats, assumptions, and security policies.

There are also external systems in CCMODE Tools which support:

- assigning a version number to files and documents – Subversion (SVN) application;
- modeling, development and analyses which are made with the use of UML (Unified Modeling Language) – Enterprise Architect (EA);
- flaws reporting and flaws remediation – Redmine;
- management and planning of TOE tests – TestLink.

The evaluation module is used to verify the development environment against the CC requirements and to evaluate evidence documents according to the CEM methodology.

If the project of the IT product is completely configured then creating evidence documents can start by using the documents generator called GenDoc for short.

### B. Documents generator (GenDoc)

GenDoc is used for editing evidence documents based on the design patterns. In order to evaluate the TOE, an ST document and accompanying documents must be prepared. These additional documents are determined by the chosen EAL and its SAR components.

This section shows on the example of an ST document how the software tool is used for filling in the patterns. Fig. 2 depicts the example of the GenDoc window with the ST design pattern. The general structure of the patterns, context-sensitive help and data fields were described. The precise details of the security development procedure and working out the evidence documents in the context of biometric devices can be found in [23].

Every pattern in GenDoc was prepared as a tree of data fields which represent chapters, sections and subsections of the output document. The tree is based on the requirements of the given CC component. The colors of branches show which fields have to be filled in by the user (red ones), which are already filled (black ones), and which are without any data (brown ones). The gray colored fields are automatically filled in with the information taken from the knowledge base and external modules: EA, EMT, SVN, TestLink.

In order to complete the document, the user must follow all the tree branches and find out which fields have to be completed. Every field has its own context-sensitive help which gives necessary guidelines and hints about the information to be delivered.

### C. Context-sensitive help

Preparation of data fields content can be facilitated by context-sensitive help. This help is accessible from the main window of GenDoc by the link “Help – access to knowledge base”. There were five types of help applied: “ready to use” – it comprises a text which is ready to use by the user without the necessity to change any information in it; “Common Criteria help” – it comprises all the information and requirements taken from CC; “hints” – these are interpretations, tips and guidelines; “example” – it is an optional text which illustrates what kind of data can be written in the given field; “data source” – it indicates an external system which is the

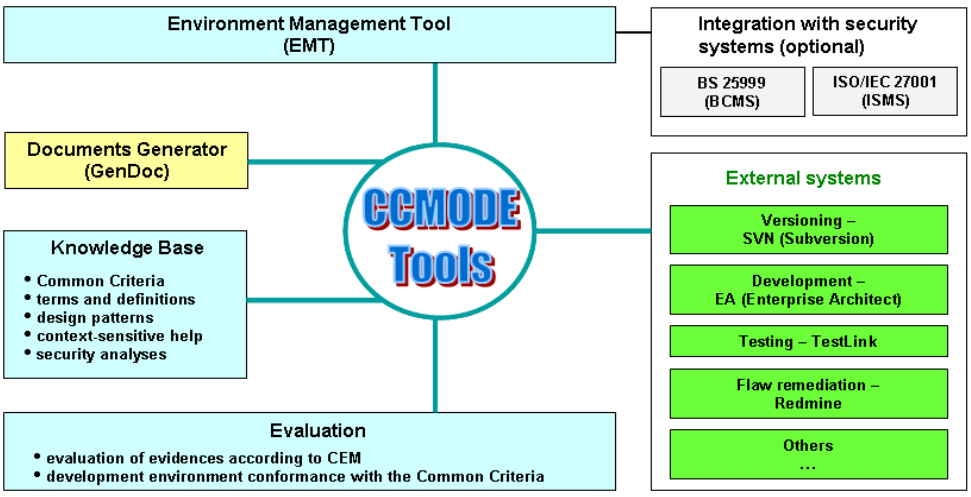


Fig. 1. The general model of the CCMODE Tools system

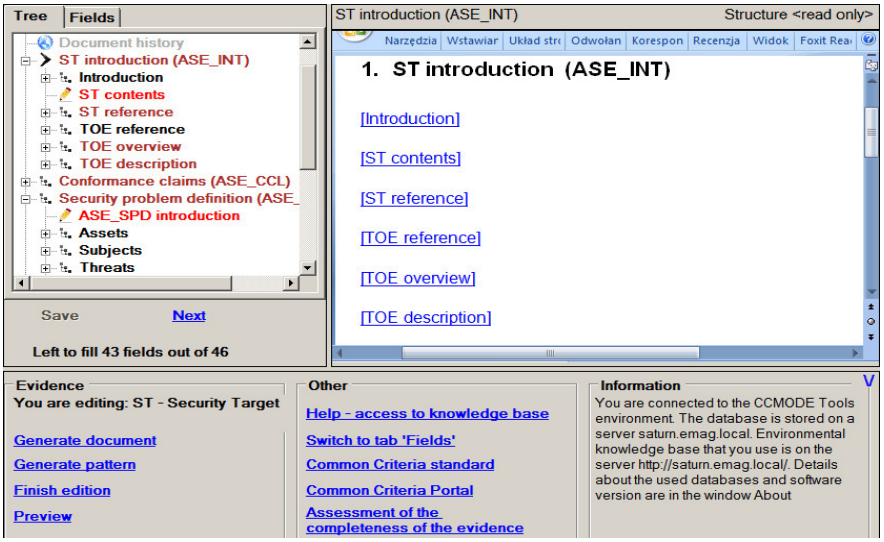

Fig. 2. The GenDoc window with the ST design pattern

source of data for the given data field. All the design patterns implemented into the CCMODE Tools system have similar representations. They contain data fields with precise instructions how to generate a complete evidence document.

At every stage of the edition process the data fields can be reviewed and checked. Verification of the document can be done with the use of the evaluation module as it is described in the next section.

### *D. Generation and revision of evidence documents*

After completing all the information in the pattern, the user can verify the output document by using an evaluation module which is a part of the EMT system (Fig. 3). This module enables to check the document according to the CEM evaluation methodology.

In general, the methodology specifies elements which describe evaluation tasks to be done by the evaluator. These tasks give precise information how each security assurance component should be checked. Every task consists of a set of questions referring to the content and form of the evidence document. These questions are grouped in the so called work units. The answers lead to work units verdicts which can have one of three possible states: pass, fail or inconclusive. Each verdict needs short justification. All verdicts are initially inconclusive and remain so until either a pass or fail verdict is assigned. Verification of the evidence document is positive when all the verdicts are passed.

The evaluation module consists of work units with their detailed descriptions and has an answer form with a built-in justification field as it is depicted in Fig. 3. The developer has to answer all these questions which pertain to the verified document.

The enhanced version of the evaluation module was applied in GenDoc where the work units are directly connected to the relevant chapters and subsections of the evidence document in order to make the verification process easier and faster. This way the developer can see the content to be checked and the relevant work unit in one GenDoc window.

After verification, the complete document can be generated as an MS Word document and saved in the SVN repository. This document can be edited in a standard MS Word editor. The document has a fixed structure with chapters, sections and subsections. It contains also footnotes with hints and guidelines.

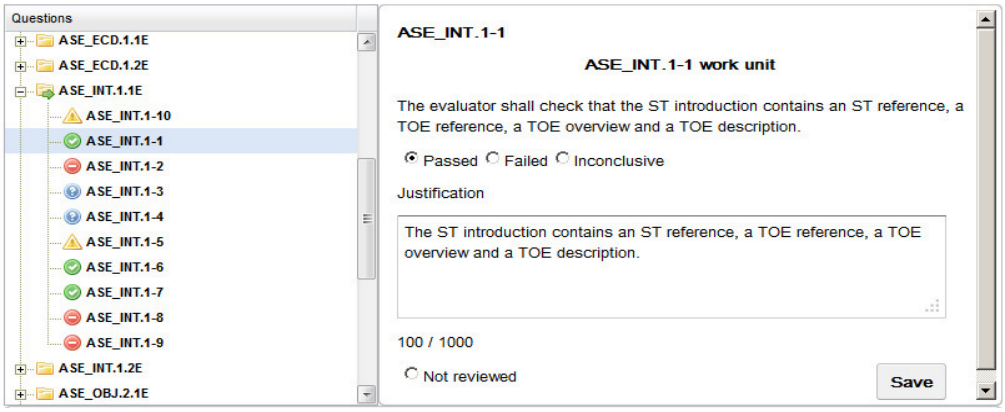

Fig. 3. The evaluation module of the EMT system

## VI. Conclusions

This work presented software implementation of the design patterns which were worked out in the CCMODE project. The patterns were positively checked and validated by developers of IT secure products but at the same time they demanded for some automation features. This is why the CCMODE Tools system and the documents generator GenDoc were developed in order to support preparing of evidence documentation.

The proposed solution based on the patterns-based approach improves the IT security development process. It overcomes a lack of knowledge and experience of the user. The context-sensitive help connected to every field of the pattern allows the developers to concentrate only on writing the proper content.

The software tool facilitates and speeds up the IT security development process and improves the quality of evidences, which become more consistent and include all details required by the CC assurance requirements. The CCMODE Tools system gives a great chance to prepare all documentation for successful Common Criteria evaluation process. Additional self-evaluation and verification enhanced functions are also implemented in the GenDoc tool. These offer a practical way of documents evaluation according to the CEM methodology but it will be the topic of the next paper.

Future work will be focused on building a standalone, independent GenDoc application which could work without the EMT framework. It is demanded by some developers who elaborate only evidence documentation. In the future work it will also be considered to adapt the documents generator to produce documents according to different typesetting systems.

## References


[1] *Common Criteria for Information Technology Security Evaluation (Version 3.1, Revision 4) Part 1: Introduction and general model (ISO/IEC 15408-1), September 2012*.

[2] *Common Criteria for Information Technology Security Evaluation (Version 3.1, Revision 4) Part 2: Part 2: Security functional requirements (ISO/IEC 15408-2), September 2012*.

[3] *Common Criteria for Information Technology Security Evaluation (Version 3.1, Revision 4) Part 3: Part 3: Security assurance requirements (ISO/IEC 15408-3), September 2012*.

[4] Common Methodology for Information Technology Security Evaluation (Version 3.1, Revision 4) Evaluation Methodology, September 2012.

[5] W. Jackson "Under attack." (GCN), August 10, 2007.

[6] A. Bialas, "Semiformal Common Criteria compliant IT security development framework." Studia Informatica 2008, 29, No. 2B(77); Silesian University of Technology Press Gliwice, Poland.

[7] D. Rogowski, P. Nowak, "Pattern based support for Site Certification." *W. Zamojski et. al. (Eds.): Complex Systems and Dependability*, AISC Vol. 170, pp. 179-193, Springer-Verlag Berlin Heidelberg 2012.

[8] I. Kane, "Automated tools for supporting CC design evidence," 9th International Common Criteria Conference, Jeju, South Korea, 2008.

[9] 13th International Common Criteria Conference, Paris, France, 2012.

[10] *ISO/IEC TR 15446: 2009 "Information technology – security techniques – guide for the production of Protection Profiles and Security Targets"*.

[11] "The PP/ST guide," Version 1, Revision 6.2, BSI, August 2007.

[12] "Guidelines for developer documentation according to Common Criteria Version 3.1," BSI, 2007.

[13] "Guidelines for evaluation reports according to Common Criteria Version 3.1," Version 2.00 for CCv3.1 rev. 3, BSI, 2010.

[14] Higaki, Wesley Hisao: *"Successful Common Criteria evaluations. A practical guide for vendors"*, 2010.

[15] D. Horie, K. Yajima, N. Azimah, Y. Goto, J. Cheng, "GEST: A generator of ISO/IEC15408 Security Target templates". *Computer and Information Science* 2009, pp 149-158.

[16] www.trusted-labs.com: Accessed May 2013.

[17] A. Bialas, "Specification means definition for the Common Criteria compliant development process – an ontological approach." In: *Zamojski W., Mazurkiewicz J., Sugier J., Walkowiak T., Kacprzyk J. (Eds.): Complex Systems and Dependability*; AISC Vol. 170, Springer-Verlag: ISBN 978-3-642-30662-4, pp. 37-54, 2012.

[18] A. Bialas, "Security-related design patterns for intelligent sensors requiring measurable assurance." *Electrical Review*, ISSN 0033-2097, vol. 85 (R.85), Number 7/2009, pp. 92-99, Sigma-NOT, Warsaw.

[19] A. Bialas, "Common Criteria related security design patterns for intelligent sensors – knowledge engineering-based implementation. In: *SENSORS*, Volume 11, Issue 8, Pages: 8085-8114, DOI: 10.3390/s110808085, 2011.

[20] A. Bialas, "Patterns improving the Common Criteria compliant IT security development process." In: *Zamojski W, Kacprzyk J, Mazurkiewicz J, Sugier J, Walkowiak T (Eds.) Dependable Computer Systems*; AISC, Vol. 97, pp. 1-16 Springer-Verlag: Berlin Heidelberg, 2011.

[21] J. Baginski, A. Bialas "Validation of the software supporting information security and business continuity management processes." In: *Zamojski W., Mazurkiewicz J., Sugier J., Walkowiak T., Kacprzyk J. (Eds.): Complex Systems and Dependability*; AISC, Vol. 170, Springer-Verlag: Heidelberg, 2012, pp. 1-18.

[22] A. Białas, "Computer support for the development process of security-enhanced IT products," Original Polish title: Komputerowe wspomaganie procesu rozwoju produktów informatycznych o podwyższonych wymaganiach bezpieczeństwa. Wydawnictwo Instytutu Technik Innowacyjnych EMAG, financed by UE POIG 1.3.1, Katowice ISBN 978-83-932737-8-2, , 2012.

[23] A. Białas, "How to develop a biometric system with claimed assurance," Proceedings of the 2013 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 775–780, ISBN 978-1-4673-4471-5 (Web), IEEE Catalog Number: CFP1385N-ART (Web).