Paper

Secured Workstation to Process the Data of Different Classification Levels

Zbigniew Zieliński[a], Janusz Furtak[a], Jan Chudzikiewicz[a], Andrzej Stasiak[a], and Marek Brudka[b]

[a] *Military University of Technology, Warsaw, Poland*
[b] *FILBICO Ltd, Zielonka, Poland*

Abstract—The paper presents some of the results obtained within the ongoing project related with functional requirements and design models of secure workstation for special applications (SWSA). SWSA project is directed toward the combination of the existing hardware and software virtualization with cryptography and identification technologies to ensure the security of multilevel classified data by means of some formal methods. In the paper the requirements for SWSA, its hardware and software architecture, selected security solution for data processing and utilized approach to designing secure software are presented. The novel method for secure software design employs dedicated tools to verify the confidentiality and the integrity of data using Unified Modeling Language (UML) models. In general, the UML security models are embedded in and simulated with the system architecture models, thus the security problems in SWSA can be detected early during the software design. The application of UML topology models enables also to verify the fundamental requirement for MLS systems, namely the hardware isolation of subjects from different security domains.

Keywords—cryptographic protection, multilevel security, software design, UML, virtualization.

1. Introduction

The issue of building a reliable Specialized Computer Systems (SCS), which are processing data with different levels of sensitivity becomes particularly topical, especially in regard to the SCS applications in government institutions, military or financial. The problem of processing information with different levels of sensitivity has been extensively studied since the early 70s of the twentieth century [1]–[3]. Formal base of multilevel security (MLS) are presented in the work of Bell-LaPadula (B-LP) [2]. In the computerized system, which uses a multilevel security, it is necessary to determine users authorization (so-called security clearance) who work with classified information in accordance with the requirements of the missions' tasks (the rule of "necessary knowledge") and the classification of information, due to the required level of protection.

To ensure the confidentiality and integrity of information, there are often used models of B-LP, Biba [4], Clark-Wilson [5] which provide mandatory access control (MAC) entity (called subject) to the resource (called object). The mandatory access control of any entity (this can be a process) to resource (e.g., data file, the communication channel, etc.) is assigned to a security context. In order to determine entitlements in systems using MAC are designed labels, which contain the security context in particular pairs: *<sensitivity level, information category>*. On the set of labels of protected data partial order relation is defined, and to subjects and objects must be used invariable rules [3], [4], [6], among others, a rule prohibiting "writing down", a rule prohibiting "reading up". It should be noted that implementing the systems and networks' set of rules (that is, building a reliable system based solely on operating systems with multilevel protection of information) in the computer is difficult and expensive. This is mainly because of difficulties to build a reliable reference monitor and the difficulty of ensuring that in the system will not be the "leak" of sensitive information due to the possibility of the so-called covert communication channels in operating system [7].

Another approach to the construction of a centralized (ie, no distributed) computer system with multi-level security is to develop software in the virtualization technology [8]–[10] for the separation of independent security domains, called the Multiple Independent Levels of Security. Such software should allow for the simultaneous launch of several specific instances of operating systems on one PC (such as a workstation or server) designed to process data of different classification levels (e.g., public and proprietary), or to process the data in different systems for which there need for separation of data.

This approach has become today entirely possible thanks to the availability of solutions with virtualization hardware support in modern Intel and AMD processors, and developed software packages (COTS type) for virtualization. Now, widely used are the extension of the x86 architecture, designed to support hardware virtualization [8]–[10] as Intel Virtualization Technology in particular VTx, VTD for x86 processors, VTi for Intel IA-64 (Itanium), and AMD Virtualization (AMD-V) for 64-bit x86 processors from AMD. These technologies also allow (in addition to hardware support emulation of the virtual machine) for building a trusted environment in which a separate virtual machines (which are separate security domains) are performed in separated hardware partitions. The implementing of this type of design of MLS system requires the integration of available virtualization technology (software and hardware), application of formal methods for both ensuring and control of the confidentiality and integrity of data, and techniques for user authentication. A natural way to build such systems is component approach, which assumes the use of

ready and available hardware components and software, in particular virtualization packages available as open source like Xen [11] and KVM [12].
The paper presents the requirements for a Secure Workstation for Special Applications (SWSA), its hardware and software architecture, selected security solution for data processing and utilized approach to designing secure software. The developed method of manufacturing the type systems of MLS, which is defined as a Model Driven Multilevel Security (MDmls) method, organizes the process of producing the SCS of MLS type and is derived from the concept of Model Driven Architecture (MDA) [13], [14] and Model Driven Development (MDD) [15], [16]. A similar approach to build secure software is presented in [15], but it does not include multilevel protection issues. The integration of security models with models of systems described in UML enables the simulation which allows to verify the effectiveness of security the designed software of SCS type MLS at the stage of modeling.

2. SWSA Requirements

The technical solutions should be designed and prepared for strictly defined applications. This paradigm is particularly important when considering the high assurance equipment. The invalid definition of SWSA designation could lead to a significant increase in costs, realization time and the complexity of security functions, and as a result to reduce the chances of achieving the adequate quality, reliability and security assurance of the product. The first task in the project was therefore to consider the technology limits imposed by legal requirements and then to select the suitable usage scenarios.

2.1. SWSA Usage Scenarios

The most important discriminating factors for SWSA usage scenarios were adopted ad hoc, just to make the legal analysis of usage scenarios more thorough and systematic:

- differing classification levels of information which is processed within individual virtual machines: *single-level*, *multi-level*, and *international multi-level*, while the latter is not considered in the subsequent analysis;
- the number of users accessing SWSA categorized as *user-less* and *multi-user*; in user-less applications the Administrator (ADM) and the Security Officer (SO) only have access to the system, while in multi-user applications there are some additional users of the workstation with access rights other than ADM and SO;
- the connectivity of SWSA, which is recognized as: *stand-alone*, *local-area* and *wide-area*; the stand-alone SWSA has no access to any ICT networks; the locally connected SWSA accesses a local area network within a single security zone, while the wide-area connectivity implies that SWSA is connected to a wide area networks and may access multiple ICT networks located in different security zones.

The combinations of the aforementioned factors lead to different legal implications for related classes of SWSA usage scenarios. The analysis of these combinations allowed to arrange them in order of the increasing complexity of the most important implementations, or to be more specific, the anticipated complexity of obtaining the security approval during the accreditation:

- *single-level*, *multi-user* and *wide-area* class of usage scenarios. The workstation in these scenarios is connected to several networks processing information of the same security classification, but differing categories; an example of such application is the mutual access of SWSA to the networks of R&D and accounting departments;
- *multi-level*, *multi-user*, and *stand-alone* class of usage scenarios; example of such applications may be the stand-alone trusted workstation in classified information storage facility;
- *multi-level*, *user-less*, and *wide-area* class of usage scenarios, in which the SWSA could be the base platform for ICT security assurance solutions, e.g., to transfer the data between systems processing the information of differing levels of classification;
- *multi-level*, multi-user and *wide-area* class of usage scenarios; the security requirements and limitations for such applications are the most demanding when compared to any other class of applications; in general, these requirements and limitations regard the challenging threats for the confidentiality of information in multi-level, multi-user and distributed environments.

2.2. Legal Regulations and SWSA

The main conclusion of a survey on national regulations and guidelines in ICT security is the observation that these legal documents apply mainly to the "hardware" level of ICT, and do not contain any specific requirements for the virtualization. The lack of necessary regulations does not imply, however, the application of virtualization in the classified information systems is forbidden due to the well-known security principle of "what is not allowed – it is forbidden". While some changes of the legal status could help to implement such solutions, there is always the opportunity to obtain the approval for such applications within the system accreditation procedures. The next statements are the fundamental security terms and requirements for SWSA identified during the analysis of the Polish Classified Information Protection Act [17], regulations and security authorities guidelines.
The most important requirement is that the SWSA must ensure the protection adequate to the highest level of classification of the information processed within VMs. It is

assumed that due to the anticipated classification level such a security assurance of SWSA should to be evaluated on at least 4th level (EAL4) in accordance to Common Criteria methodology. It was also found that at least the compartmented security mode is adequate for applications demanding the security functions of SWSA. As a result, SWSA should provide the technical support for strict mandatory access control (MAC) policies.

In the *wide-area* applications, in which the distinct VMs are connected to local area networks of cooperating entities, SWSA should comply with the security requirements for system interconnection. In particular, SWSA must provide security measures to absolutely protect the confidentiality of the information. These security functions should be accompanied by proof of a controlled separation of SOS including the covert channels analysis.

The *wide-area* application imposes some specific requirements on the formal labeling and registration of each workstation and individual VMs. These terms are in a sense analogous to the deployment of the remote IT terminal within the security zone, which is not controlled by the system owner. The host system of SWSA comprising of the computer, hyper-visor and the host operating system should be labeled and registered as a part of ICT system in the deployment place. VMs, which are remotely attached to other ICT systems via wide area connections, should be labeled and registered as agreed between the cooperating entities and the system owners. The agreements may vary with respect to the ownership, administration duties and responsibilities, and even liabilities regarding VM. SWSA should therefore provide some technical and operational support regarding the registration and labeling of virtual machines as well as their backups.

The Classified Information Protection Act imposes also the obligations to implement the security measures to protect ICT equipments against the compromising electromagnetic emanations. In general, these obligations apply to the hardware part of the workstation, namely chassis, signal and power supply lines. In particular, there are some specific requirements on the separation of the signal and supply lines belonging to either the unclassified (BLACK) or classified (RED) parts of ICT system. However, these conditions do not imply any separation requirements for the parts of SWSA which process the classified information of different security level. It is therefore assumed that SWSA should host only either the RED or the BLACK VMs.

2.3. Specification of Functional Requirements

In the SWSA environment can distinguish three types of actors: the system administrator, security officer (hereinafter SO), SOS user (hereinafter user) (Fig. 1).

Security officer with the administrator and others are developing special security requirements of the system (SSRS), and safe operation procedures (SOP). The SSRS is identifing levels of security virtual machines installed in SWSA and permissions for actors (users). SL involves clause of

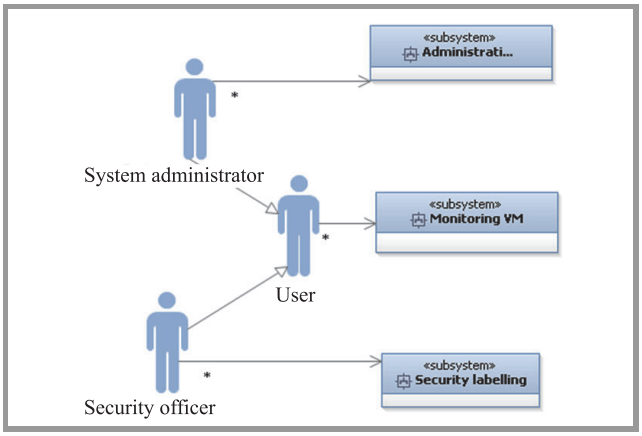

Fig. 1. General scheme of the basic elements of SWSA architecture.

information authorized to process and its set of information categories (range). Each level is described (according to Bell-LaPadula model) by a pair $<k, c>$ where $k \in K$ is the clause of information ($K = \{$public, proprietary, confidential, secret$\}$), and $c \in C$ is a subset of the categories of information $c = \{c_1, c_2, \ldots, c_L\}$. For example, $C = \{PD, GR, OS, DP, \ldots\}$, where the symbols PD, GR, OS, and DP denote the personal data, guidance resources, the operational situation, the data for purposes.

The clauses are ordered (from minor to major), $\forall_{i=(1,\ldots,4)}\, k_i \leq k_{i+1}$, but the categories are not. Security levels can be compared. For example, $SL_a = <k_a, C_a>$ and $SL_b = <k_b, C_b>$, if the following conditions: $k_b \leq k_a$ and $C_b \subseteq C_a$, then $SL_b \leq SL_a$ (SL_a level is higher or equal than the SL_b). Let $SL_b = <confidential, \{PD, GR\}>$, and $SL_a = <secret, \{PD, GR, OS\}>$, then we have a $SL_b \leq SL_a$, because according to satisfy the following: $confidential < secret$ and $\{PD, GR\} \subseteq \{PD, GR, OS\}$. Please note also that not all pairs of security levels are comparable. This leads to the use of the concept of lattice of security levels.

The security system according to Bell-LaPadula model is satisfied if the following axioms are preserved: security simple, stars, stability, security discretionary, non-availability of inactive object, the independence of the initial state. These axioms have been adopted in all models using mandatory access control to information. The fulfillment of these axioms ensures that classified information in the system will not be available for those who did not receive proper authorization. SO defines security attributes of SOS_k $(k = 1, 2, \ldots, n)$ existing in the SWSA, such as their clauses and classes of applications, manages the database of users and their security credentials, identifying opportunities to access resources for each of the domains. Access permissions to the domains are determined by security labels. Security officer gets access to the labels management and control of their allocation in the system through Virtual Machine Monitor (VMM).

The administrator performs backups of host machines and virtual machines, creates a new account for SSO users,

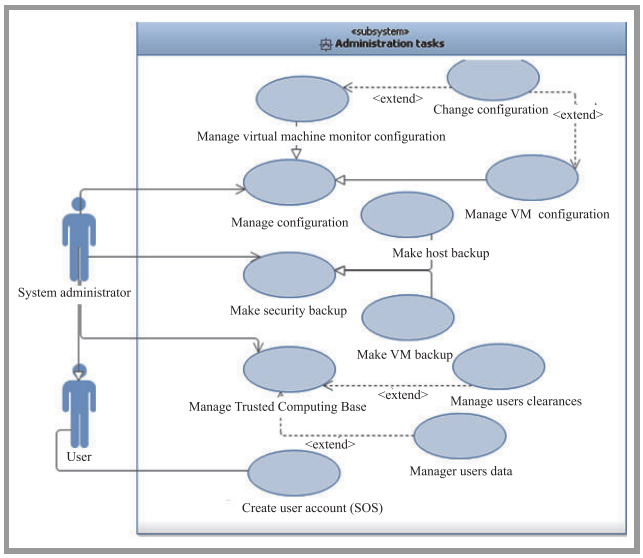

Fig. 2. Functional requirements for administration subsystem of SWSA.

creates and modifies the configuration of virtual machines (Fig. 2). In the model of requirements for VMM (Fig. 1), the user limits his task to run the SSO.

3. Architecture SWSA

It is essential for creating complex computer systems plays an architecture solution. With regard to the SWSA to be particularly important to recognize the hardware and software elements of the architecture due to their significant impact on the security of the system.

3.1. Hardware Architecture

It is assumed the use of components that enable hardware support for security technology and hardware virtualization support. To isolate the separate domains containing isolated environments implementing Trusted Execution Technology (TXT) will be used, while an important role in ensuring the integrity of the SWSA will play a Trusted Platform Module (TPM) that allows the secure creation and storage encryption keys.

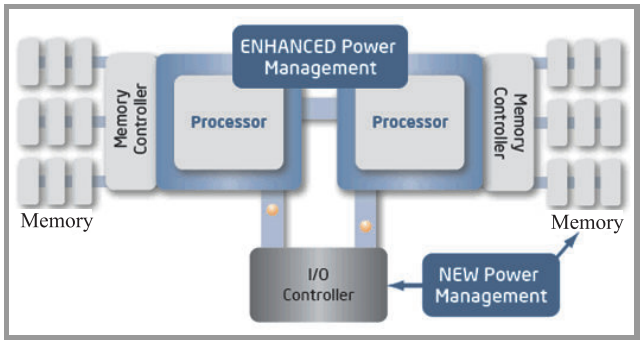

Fig. 3. Block diagram of a system based on the Xeon processor 5600 series.

Because of the applied approach (using ready components), hardware design limits itself to the configuration that describes the characteristics of the applied solution. Proposed hardware architecture of SWSA is based on a machine with an Intel Westmere with dual-processor Xeon E5630 model (Fig. 3), each of which contains the four processor cores. This architecture gives the possibility of sharing the responsibilities between the various hardware components, allows for the separation of the flow of information within the hardware and allows the separation of partitions with distinct security domains. Block diagram of a system based on the Xeon processor 5600 series is shown in Fig. 3. When designing the hardware architecture used UML with using topological models which construction is supported by a CASE environment – Rational Software Architect (RSA).

3.2. Software Architecture

In the architecture SWSA can be distinguished the following elements: the trusted system platform (TSP) and executable special versions of operating systems SOS. The general scheme of architecture SWSA is shown in Fig. 4.

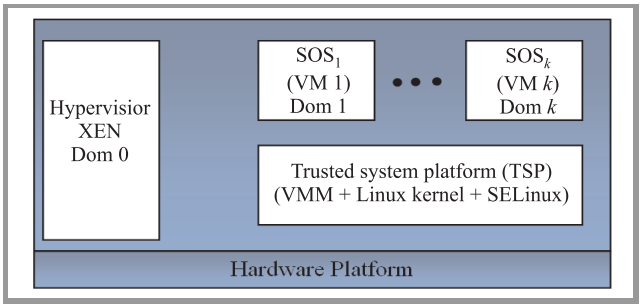

Fig. 4. General scheme of the basic elements of SWSA architecture.

The essential elements of an architecture SWSA are the SSP and virtual machines VM_i which we write in the form of the formula: $SWSA := SSP + \{VM_i\}$, where $i = \{1, \ldots n\}$. SSP consists of: kernel of Linux operating system, its extension in the form of a component (SELinux), and monitor virtual machines (*VMM*), *TSP = Linux_Kernel + SELinux + + VMM*. *TSP* allows you to run and supervise activities of specialized operating systems SOS_i $(i = 1, 2, \ldots n)$ and acting on their environment, application programs $\{AP\}$, forming a virtual machine, $VM_i := SOS_i + \{AP\}_i$.

The VMM is a key component of the SWSA, responsible for running virtual machines in accordance with defined security rules (using the hardware support) and their switching to ensure the separation of resources. It was assumed that the proposed VMM software should make it possible to simultaneously launch several (of many possible) instances of special versions of operating systems SOS_i on a single computer with the provision of: access control, separation of resources, cryptographic protection, and strict

control of data flow. The number of instances of virtual machines that is run, depends on the configuration of the station, in particular on the number of physical processors and cores. For example (Fig. 4), SSP supervises n virtual machines, among which operate both VM_1 and VM_2, and accordingly, on the VM_1 is active $\{AP\}_1$ (which is represented as $VM_1 \rightarrow \{AP\}_1$), on VM_2 is active $\{AP\}_2$ (which is represented as $VM_2 \rightarrow \{AP\}_2$). The VMM manages access to both virtual machines SOS_i, as well as to hardware resources (physical and virtualized).

The project also assumes that the instance of a special version of the operating system (SOS) working within a virtual machine (VM) is a separate Security domain (SD), $VM_i\ [SSO_i] = SD_i$. In each of the domains the processing of the data qualified to different security levels is allowed. Figure 4 shows two security domains, and each of them associated with one virtual machine.

It is worth noting that both the operating system kernel, as well as special versions of operating systems in terms of the project are ready components and their design will be limited only to the specifications of their interfaces and configuration descriptions. Interfaces were described in UML, and the configurations on the topological diagrams. In this area, the CASE tools (RSA) were used.

4. Cryptographic Protection of SWSA

Even in the simplest systems and applications, there are many places where the potential attack is possible, and their number is limited only by the inventiveness of the attacker. There are three basic areas in which information is exposed to capture:

- when you enter (e.g., keyboard);
- during transmission (e.g, via a local network or the Internet);
- when writing (e.g., on fixed and removable media).

In the area of interest of presented data protection solution, the SWSA belongs to the third area, including security of data stored on fixed and removable storage media.

The elaborated solution is assumed, among other things, the exchange of data between the internal, to the operating system, medium (ie hard drive), and external media (e.g., hard drive or Flash RAM) connected to the system via USB. However, the data exchanging between external media (e.g., Flash RAM) which are connected to the SWSA by USB is not possible. The process of securing the exchange of data using removable media should satisfy the following functional requirements:

- should be implemented in a manner transparent to the user;
- should not cause any noticeable to the user loads of the operating system;
- should not have a significant impact on the speed of read and write data onto data media;
- should allow to perform any operation allowed for data storage media, such as volume, surface checking for errors, and defragment the disk-based data.

These requirements force the use of the process of data security, a dedicated module (driver) for the operating system running at the kernel level. Schematic diagram of the developed solution is shown in Fig. 5. A detailed description of the method of securing removable media is given in [18].

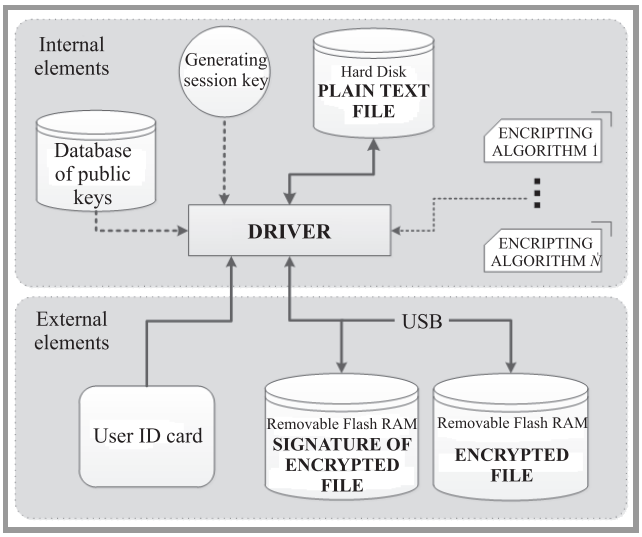

Fig. 5. Schematic diagram of securing of data stored on an external drive.

It is assumed that the elements involved in the process of securing stored data are divided into two groups: the internal and external components. The division has been completed, taking as its criterion, the relationship between the elements and the operating system installed on computer. The internal components (Fig. 5) includes both the hardware in a computer's hard drive, and software modules: the driver, `.DLL` library that provides the functionality of the implemented encryption algorithms, module of session key generation, and a database of public keys of users. External components (Fig. 5) connected to a computer via USB include hardware components in the form of Flash RAM, and the user identification card in the form of a smart card.

The process of writing (encryption) of data transferred to external media requires the user to identify the recipient of such a data. The process of data recording is initialized by the user (operator) currently logged on the system. The logged user may also be the recipient of the data. During the reading operations (decryption) the file, the operator is the recipient. For each of the saved file, the signature

is created that contains the information needed to decrypt the file. A signature is cryptographically protected, and could be read only by the recipient of the file.

5. The Method of SWSA Software Design

The key problem of SWSA software design boils down to building the trusted system platform (TSP) which includes: operating system kernel, virtual machines manager (VMM), and running special versions of operating systems. It is worth noting that both the operating system kernel, and special versions of operating systems in terms of the carried out project are ready components, and the project will be limited only to the specifications of their interfaces and configuration descriptions. Interfaces would be described in UML and the configurations on topological diagrams.

The essential complexity of software design is thus reduced to the construction of VMM virtual machines manager (Fig. 4), which will be responsible for its own implementation of the Xen hypervisor virtualization component, but it is worth noting that the work includes the implementation of its own unique solutions in this scope, in particular, it provides multilevel security policies.

In the process of development of VMM software, a new method of software design of MLS-type systems called MDmls was proposed [19] which is based on MDD (Model Driven Development) approach [13]. The method intended for designing specialized MLS-type systems contains, in particular, its basic processes, domain languages used, stages and development environment. The essence of the MDmls method is the integration of MLS security models with system design models expressed in UML-based language. Such integrated models with both a concrete notation and abstract syntax are called security design models [20].

In the MDmls method, the activities concerning of domain modeling languages DSM are essential. The construction of a new class of Domain Specific Language requires a metamodel created, because only on this basis the profile can be defined. The metamodel formalizes the structure of models, as well as scenarios, which represent possible instances of SWSA.

From the perspective of project management, the transition to the implementation stage takes place after completing modeling, which we build the next release of "tested models" in an incremental and agile way. Implementation, however, is carried out, but only after delivering the final version of the system model, and considerably makes up the result of automatic transformation of models into system code and descriptions of the required configuration. Therefore, the method assumes that all developed, in accordance with MDA, models are combined with transformations: speed manual or automatic (model-to-model [M2M], model-to-text [M2T]).

In this scope of design process, it is proposed to use the extended UML language with the so-called topological models [16], [21]. The creation of these models is supported by the CASE environment – Rational Software Architect (RSA). The IBM RSA ver. 8.0 extended environment was used with the Rational Software Architect Simulation Toolkit, with support for UML Action Language (UAL), which provides a subset of the specifications described in OMG with technologies fUML and ALF (Fig. 6).

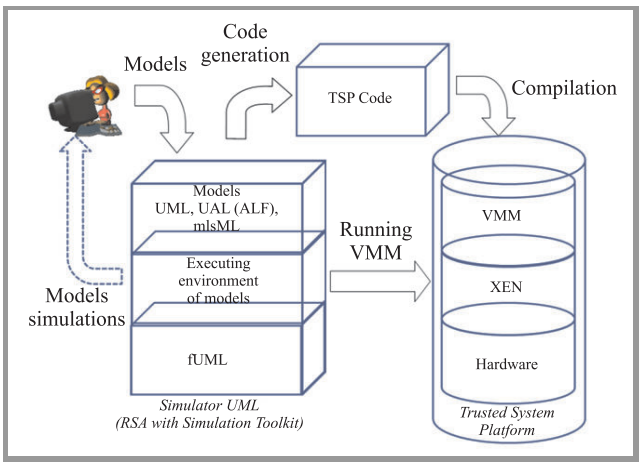

Fig. 6. The outline of environment for development of SWSA software [19].

The description of methods on how to validate the design solutions, specification of threat scenarios, and verification of the models in the RSA environment the subject of another publication.

Thanks to the integration of security models with the MLS-type SCS models (described in UML), in the proposed method of designing MDmls, the possibility of simulating models is obtained, which allows verification of many security problems of the designed system at the modeling stage. Using the model topology also allows the (physical) study of separating data processing processes belonging to different domains of security, which is one of the elements of verifying MLS-type systems. Additionally, the topology model enables defining the SWSA configuration, and then validate the rules concerning the exhaustion of station physical resources (resulting from its current configuration).

6. Summary

The main goal of the project that is to develop a secure environment for processing data of different classification levels on the same physical machine is achieved by integration of existing hardware and software virtualization, cryptography and identification technologies to ensure the security of multilevel classified data by means of some formal methods and components approach to provide different virtual machines with either Linux or Windows systems for each security level. The SWSA project is currently in the validation phase and its results are quite promising.

Acknowledgements

This work was supported by The National Center for Research and Development, Project OR00014011.

References

[1] J. P. Anderson, "Computer Security Technology Planning Study", vol. II, ESD-TR-73-51. Electronic System Division, Air Force System Command, L. G. Hanscom Field, Bedford, MA 01730, USA, Oct. 1972.

[2] D. E. Bell and L. J. La Padula, "Secure computer system: unified exposition and multics interpretation", ESD-TR-75-306, Bedford, MA: ESD/AFSC, Hanscom AFB [Online]. Available: http://csrc.nist.gov/publications/history/bell76.pdf

[3] D. E. Bell, "Looking back at the Bell-La Padula model", in *Proc. 21st Ann. Comp. Secur. Appl. Conf. ACSAC 2005*, Tucson, AZ, USA, 2005, pp. 337–351.

[4] K. J. Biba, "Integrity Considerations for secure computer systems", Tech. Rep. MTR-3153, MITRE Corporation, Bedford, Massachusetts, USA, 1975.

[5] D. Clark and D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", in *Proc. IEEE Symp. Secur. Priv. S&P 1987*, Oakland, California, USA, 1987, pp. 184–194.

[6] M. Brudka and J. Furtak, "Ponad barierami – łączenie sieci o różnych klauzulach", *Biuletyn IAiR*, no. 26, 2009 (in Polish).

[7] R. Smith, "Cost profile of a highly assured, secure operating system", *ACM Trans. Inform. Sys. Secur.*, vol. 4, no. 1, 2001, pp. 72–101.

[8] J. S. Robin and C. E. Irvine, "Analysis of the Intel Pentium's ability to support a secure virtual machine monitor", in *Proc. 9th USENIX Secur. Symp.*, Denver, Colorado, USA, 2000.

[9] C. E. Irvine *et al.*, "Overview of a high assurance architecture for distributed multilevel security", in *Proc. IEEE Sys. Man, Cybernet. Inform. Assur. Worksh.*, West Point, NY, USA, 2004.

[10] D. Kleidermacher, "Methods and applications of system virtualization using Intel® virtualization technology(Intel® VT)", *Intel® Technol. J.*, vol. 13, iss. 01, March 2009.

[11] P. Barhamet *et al.*, "Xen and the art of virtualization", University of Cambridge Computer Laboratory, CGF Brussels, 2004.

[12] A. Kivity *et al.*, "KVM: the Linux virtual machine monitor", in *Proc. Linux Symp.*, Ottawa, Ontario, Canada, 2007, pp. 225–230.

[13] D. S. Frankel, *Model Driven Architecture: Applying MDA to Enterprise Computing.* New York: Wiley, 2003.

[14] W. Dąbrowski, A. Stasiak, and M. Wolski, *Modelowanie Systemów Informatycznych w Języku UML 2.1*. Warszawa: PWN, 2007 (in Polish).

[15] T. Lodderstedt, D. A. Basin, and J. Doser, "SecureUML: a UML-based modeling language for model-driven security", in *Proc. 5th Int. Conf.*, *LNCS*, vol. 2460, 2002, pp. 426–441.

[16] "Planning deployment with the topology editor", IBM Tutorial, 2008.

[17] "Ustawa o ochronie informacji niejawnych", z dnia 5 sierpnia 2010, Dz.U. nr 182, poz. 1228 (in Polish).

[18] J. Chudzikiewicz and J. Furtak, "Cryptographic protection of removable media with a USB interface for secure workstation for special applications", *J. Telecom. Inform. Technol.*, vol. 3, pp. 22–31, 2012.

[19] Z. Zieliński, A. Stasiak, and W. Dąbrowski, "A Model Driven Method for Multilevel Security Systems Design", *Przegląd Elektrotechniczny* (*Electrical Review*), No. 2, 2012, pp. 120–125.

[20] D. Basin, M. Clavel, J. Doser, and M. Egea, "Automated Analysis of Security-Design Models", Preprint submitted to Elsevier, 2007.

[21] N. Makin, "Anatomy of a topology model in Rational Software Architect Version 7.5: Part 1: Deployment modeling", IBM, 2008.

[22] "Modeling deployment topologies", IBM Tutorial, 2008.

[23] N. Makin, "Anatomy of a topology model used in IBM Rational Software Architect Version 7.5: Part 2: Advanced concepts", IBM, 2008.

[24] S. Willard, *General Topology*. Courier Dover Publications, 2004.

[25] N. Li and J. C. Mitchell, "RT: A role-based trust management framework", in *Proc. 3rd DARPA Inform. Surviv. Conf. Exposition DISCEX III*, Washington, DC, USA, 2003, pp. 201–212.

[26] S. T. King, P. M. Chen, Y. Wang, C. Verbowski, H. J. Wang, and J. R. Lorch, "SubVirt: Implementing Malware with Virtual Machines", in *Proc. IEEE Sym. Secur. Priv. S&P 2006*, Berkeley, CA, USA, 2006.

[27] P. Ferrie, "Attacks on Virtual Machine Emulators", in *Proc. Association of Anti Virus Asia Res. Conf.*, Auckland, New Zealand, 2006.

[28] S. Mellor and M. Balcer, *Executable UML: A Foundation for Model-Driven Architecture*. Boston: Addison Wesley, 2002.

[29] M. Fowler and R. Parsons, *Domain Specific Languages*. Boston¿: Addison Wesley, 2010.

[30] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston: Addison Wesley, 2002.

[31] J. Jürjens, *Secure Systems Development with UML*. Berlin: Springer, 2010.

Zbigniew Zieliński received the M.Sc. in Computer Sciences from the Cybernetics Faculty of Military University of Technology, Warsaw, Poland in 1978, and the Ph.D. degree in Computer Systems from Military University of Telecommunication (St. Petersburg) in 1988. He is currently an Assistant Professor of Computer Systems in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology. His current research interests are in the areas of computer systems dependability, processors network diagnosis methods, fault-tolerant systems, as well as virtualization and system security.

E-mail: zzielinski@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Jan Chudzikiewicz received the M.Sc. in Computer Sciences from the Cybernetics Faculty of Military University of Technology, Warsaw, Poland in 1993, and the Ph.D. degree in Diagnosis of Computer Networks from the Cybernetics Faculty of Military University of Technology in 2001. He is currently an Assistant Professor

of Computer Systems in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology. From 1994 to 1998 he was cooperated with Industrial Institute of Electronics in domain of design of diagnostic systems for digital circuits. His current research interests are in the areas of diagnosis methods for computer systems, computer networks, and fault-tolerant systems, as well as the low-level software for the Windows systems.

E-mail: jchudzikiewicz@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Janusz Furtak received his M.Sc. from the Cybernetics Faculty of Military University of Technology, Warsaw, Poland in 1982. For eight years he was a member of the design team which developed software for command systems. Since 1990 he has been a university teacher at the Cybernetics Faculty of Military University of Technology. In 1999, he received Ph.D. degree in the field of Computer Science. Currently, he is an Assistant Professor in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology and Director of this Institute. His main areas of expertise are computer networks, network security, cyber defense and administering of network operating systems.

E-mail: jfurtak@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Andrzej Stasiak is an expert in the field of design of information systems. From years 2004–2012 he is a member of program committees of conferences on Software Engineering and Real Time Systems. From 1987 he is an Assistant Professor of Computer Systems in the Institute of Teleinformatics and Automation of Cybernetics Faculty, Military University of Technology. He gained his professional experience directing some complex IT projects.

E-mail: astasiak@wat.edu.pl
Faculty of Cybernetics
Military University of Technology
Gen. S. Kaliskiego st 2
00-908 Warsaw, Poland

Marek Brudka graduated in 1994 the Faculty of Electronics and Information Technology of Warsaw University of Technology. In 2000 he was awarded with honors a Ph.D. title for the dissertation on the intelligent robots control using neural networks and ultrasonic measurements. Since 2001 he is working for Filbico Ltd., currently as R&D manager. His professional experience comprises of the research and development as well as software development projects on robotics, command and control systems, crisis managements systems and ICT security.

E-mail: mbrudka@filbico.pl
Filbico Ltd.
Prymasa S. Wyszyńskiego st 7
05-220 Zielonka, Poland