Information & Security

Published since 1998

I. Gaidarski & P. Kutinchev
vol. 53, no. 1 (2022): 65-77
https://doi.org/10.11610/isij.5305

ISSN 0861-5160 (print), ISSN 1314-2119 (online)

Research Article

Transformation of UML Design Models of Information Security System into Agent-based Simulation Models

Ivan Gaidarski* (✉), *Pavlin Kutinchev

Institute of Information and Communication Technologies at Bulgarian Academy of Sciences, Sofia, Bulgaria, https://www.iict.bas.bg/

ABSTRACT:

The development of complex systems involves multiple participants (stakeholders) who have their own perspectives, knowledge, experience, and responsibilities that determine their requirements to the system. It is important to coordinate stakeholders and unify their requirements. We propose a method that takes into account the perspectives of all stakeholders. The framework for defining of system's problem area defines the boundaries within which the system is developed. The reference methodology for system development is based on the IEEE 1471 and 42010 standards. We use a few viewpoints: Information Security, Risk Analysis, Communication, Technological, and Information Processing. The analysis of the different perspectives allows us to construct models describing the features of the developed system. After analyzing the Information Security viewpoint, a generalized conceptual model of the system is created. The analysis of Information Processing leads to a data model. Technological point of view includes different technological approaches for the development of systems, such as object-oriented approach with UML Language for constructing the design model and agent-based modeling approach for creating a simulation model of the system. We present how the object-oriented design model described with UML can be transformed into an agent-based simulation model.

ARTICLE INFO:

RECEIVED: 28 JUNE 2022

REVISED: 25 AUG 2022

ONLINE: 20 SEP 2022

KEYWORDS:

architecture, model, agent, system, conceptual, simulation

✉ Corresponding Author: +359 88 821 1361; E-mail: ivan.gaidarski@iict.bas.bg

Introduction

The process of development of complex systems involves multiple participants (stakeholders) – each with his own perspective. Each has relevant knowledge, skills, experience, and responsibilities that determine their attitude and requirements towards the system. Modern systems use different technologies and has a variety of regulatory requirements. In the process of the system development, the different perspectives of the participants can intersect or overlap. The coordination of the interested parties and the unification of their requirements and contributions is extremely important. The environment determines the conditions under which the information security system (ISS) operates. We propose an analysis method that takes into account the perspectives of all important stakeholders involved in the system development, ensuring the comprehensiveness of the information security approaches. The problem area model matches the analysis model.

The development of ISS goes through the following stages (Fig. 1):

1. Clarifying the requirements for the ISS by analyzing the problem area.
2. Systematic analysis of requirements and construction of conceptual model of the problem area from different points of view.
3. Integrating the conceptual models created from different perspectives. This facilitates communication between the observers of the developing ISS who are associated with the respective viewpoints.
4. Creation of a ISS design model by transforming the conceptual model into an object-oriented design model.
5. Aspect-oriented transformation of a design model into an agent-based simulation model.

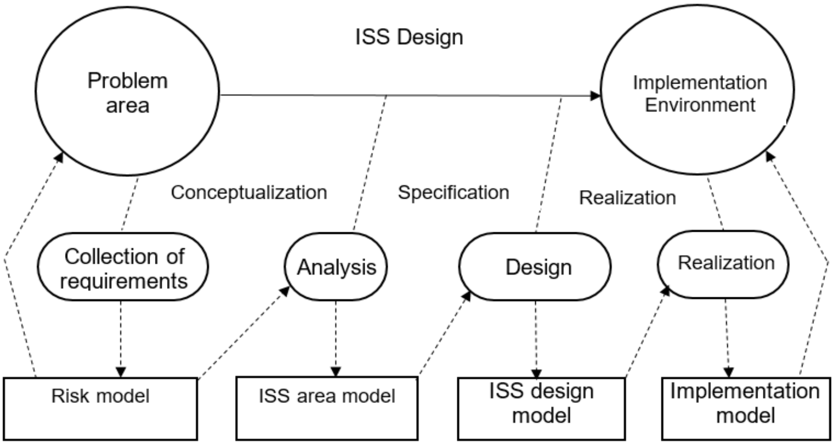

Figure 1: Systems development cycle.

Framework for Describing the System Architecture

The system framework for defining the ISS problem area and, subsequently, the system architecture defines the boundaries within which the system is developed. The reference methodology for ISS development proposed by us is based on the framework for the architectural description of software systems in the IEEE 1471 (3;4;5) and IEEE 42010 (3;5) standards.

IEEE Standard 1471 represents recommended practices regarding the creation, analysis, and maintenance of architectures of software-intensive systems and their architectural description. It offers a conceptual framework for an architectural description and defines the content of an architectural description. The IEEE 42010 standard addresses the creation, analysis, and maintenance of system architectures through architectural descriptions. A conceptual model is created, and the content of the architecture description is specified. An architecture description framework is introduced, which includes viewpoints, description languages, and common practices for describing system architecture.

The framework is formed from the multiplicity of stakeholder/observer perspectives. The architectural description initiates the creation of a design model of the ISS. It is used in the implementation of a real ISS, the design of which takes into account and unifies the requirements of different points of view in the area of interest of the ISS. The basic concepts underlying the domain analysis framework for an ISS are Environment, Stakeholder, Domain of Interest, View, Point of View, System Architecture, Architectural Description, Architectural Description Creation Framework, Architectural View, and Architectural Perspective. They define the general conceptual framework, allowing a multifaceted description of the problem area and defining the architecture of the ISS by using the capabilities of conceptual modeling (8;9;10).

Fig. 2 depicts the area of interest of the ISS, which is used to frame the analysis of the problem area of ISS.

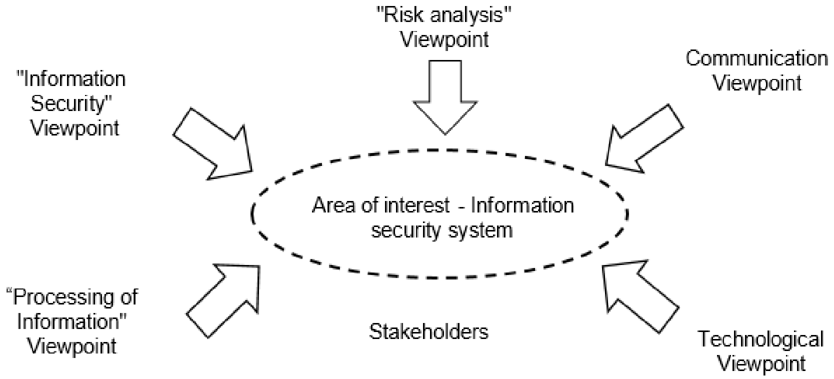
"Risk analysis" Viewpoint
"Information Security" Viewpoint
Communication Viewpoint
Area of interest - Information security system
"Processing of Information" Viewpoint
Stakeholders
Technological Viewpoint

Figure 2: Area of interest of ISS.

Our method for developing information security systems in organizations takes into account and unifies the requirements of the various elements and points of view in the area of interest of the system:

- Information Security viewpoint – includes the main concepts in information security (Threats, Vulnerabilities, Sources, Motivation, etc.), as well as the main approaches to implementing information security in organizations;
- Risk analysis viewpoint – the requirements for the ISS are determined through the risk analysis;
- Communication viewpoint – determines the way of communication, predetermining the approach to information protection.
- Technological viewpoint. This perspective includes supported platforms and technologies, as well as different approaches in information and communication technologies such as object-oriented approach and agent approach,
- Information Processing viewpoint – including the three main types of data defined according to information security – Data-in-Rest, Data-in-Motion, and Data-in-use Use).

Analysis of the different viewpoints

As a result of the analysis of the different perspectives in ISS, it is possible to construct models describing the system's features from these viewpoints. After analyzing the problem area of ISS from an "Information Security" viewpoint, using the concepts underlying the IEEE 1471 and IEEE 42010 standards, a context is provided to define a common conceptual framework for construction of conceptual models of ISS. The result of using conceptual modeling to create an analysis model of the problem area is a conceptual model that is essentially an abstraction. Each concept is considered as a separate component. Therefore, this model also represents the architecture of the ISS. The essence of our approach is the initial creation of a generalized model, and then, on its basis, a detailed model of the ISS problem area. In this way, we abstract from unnecessary details and focus on the essential features of the system. The components of the generalized model coincide with the software tasks of the designed information security system.

They reflect the relevant elements of the analysis of the field of Information Security (Fig. 3). The model consists of six components corresponding to the main concepts that represent the field of Information Security:

- "Data protection" (What do we protect?),
- "Protection of communications" (Where and What do we protect?),
- "Endpoint protection" (Where do we protect?),
- "Management and Configuration" (How do we protect?),
- "Monitoring and Analysis."

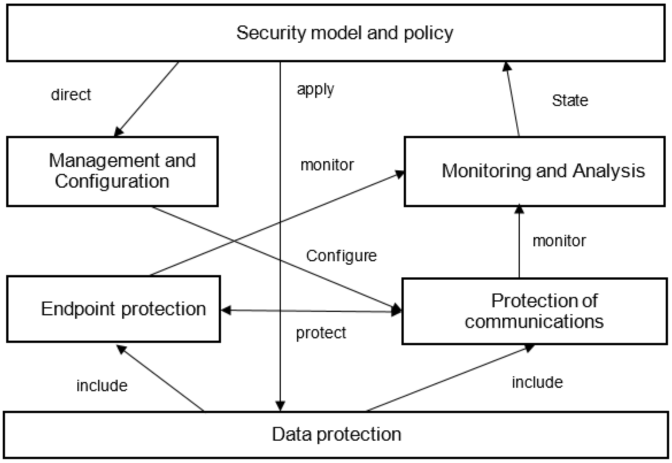

Figure 3: Generalized conceptual model of the ISS problem area.

In general, the data can be in one of the three states: “Data in motion” (communicated data, status data), “Data at rest” (storage devices, archives, network partitions) or “Data in use” (the data used or processed in applications).[1] For the formal presentation of the data in the ISS, we create a meta-model (Fig.3), which is based on the viewpoint “Processing of Information” in the area of interest of the ISS (Fig.2).[2] In order to protect the different types of data, it is necessary to implement specific approaches to information security in the main blocks of the meta-model from the viewpoint “Information Security.” Data must be protected against loss, theft, and unauthorized access or uncontrolled changes – Privacy Control, Integrity, Access Control, Isolation, and Replication.[10, 11]

Data at Rest
Data in Use
Data in Motion

Figure 4: Meta-model “Processing of Information.”

In order to take into account the requirements of all stakeholders, i.e., viewpoints, our approach allows the creation of any number of conceptual meta-models that can be combined in one system. The result is a multi-layered conceptual meta-model of the ISS which contains meta-models representing the respective viewpoint.

The analysis of the “Technological point of view” includes different technological approaches for the development of an information security system, supported platforms and technologies, as well as different approaches in information and communication technologies such as object-oriented approach and agent approach. The object-oriented approach (OOA) allows a problem to be broken down into its constituent components. Each component becomes a separate object, which contains its own data, processing methods, and certain actions. The Unified Modeling Language (UML) is widely used in OOA. The agent approach is related to the term "agent." An agent is any set that perceives the environment through sensors, processes its information, and influences the environment through actuators. Very often, the agent-based modeling approach is used to create simulation models to pre-simulate the operation of the designed ISS through available simulation environments such as NetLogo.

Object-oriented Design Model

On the basis of the proposed conceptual model, a Design Model of the system can be created, which describes the architecture (static structure) and functionality (dynamic behavior) of the ISS. An object-oriented approach and according object-oriented description language can be used for the construction of the Design Model. Such language is the Unified Modeling Language (UML), providing tools for describing, analyzing, modeling, and documenting the architecture and functionality of ISS.[6, 7]

The construction of the model is carried out by transforming the conceptual model into an object-oriented design model. The design model consists of an architectural model and a functional model, described with corresponding diagrams in UML. The static structure or architectural model of the system can be represented by the static UML structural diagrams – Class, Object, Packet, Composite Structure, Component, Deployment, and Profile diagrams. To transform the generalized model of the ISS problem area from (Fig. 2) into an OO model, we use a "Class-diagram" (Fig.5). To represent the object-oriented models of

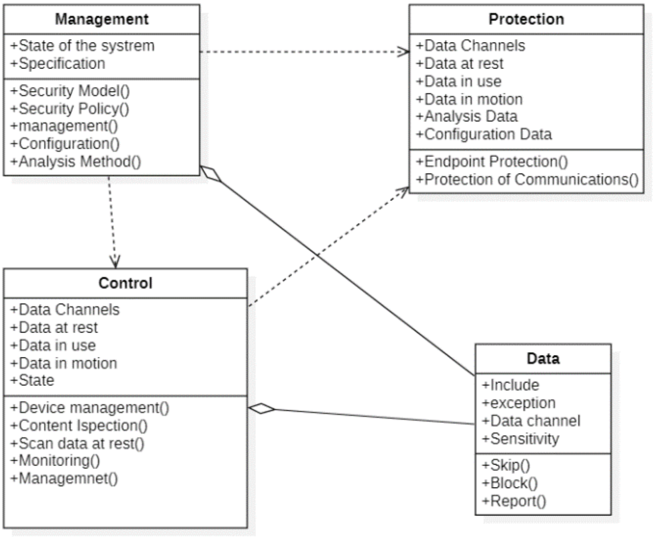

Figure 5: UML Class diagram of ISS.

the detailed models of the concepts “Endpoint protection” and “Protection of the Communications” (Fig.6) we can use a Composite structure diagram. A more detailed description of the architecture of the design model requires the use of Object Diagram and Profile Diagram from the UML arsenal.

The functional model of the ISS can be represented by dynamic UML diagrams: Behaviour Diagrams and Interaction Diagrams. Through them, different aspects of the dynamic behavior of the system and the interaction of the different elements of the system with each other or with external entities can be described. For this purpose, a dynamic analysis of the information security system is performed, and the possible interaction options are identified through the approach in Fig. 7. These options or cases can be formally described and embedded in the designed system so that it reacts to the interaction with the external subjects and the internal elements according to the goals set in its design. The set of UML diagrams describing the result of the performed dynamic analysis of the system form the ISS Functional Model. Each diagram describes individual functionalities of the system. A typical functional model consists of use case diagrams, class interaction overview diagrams, sequence diagrams, activity diagrams, and state diagrams.

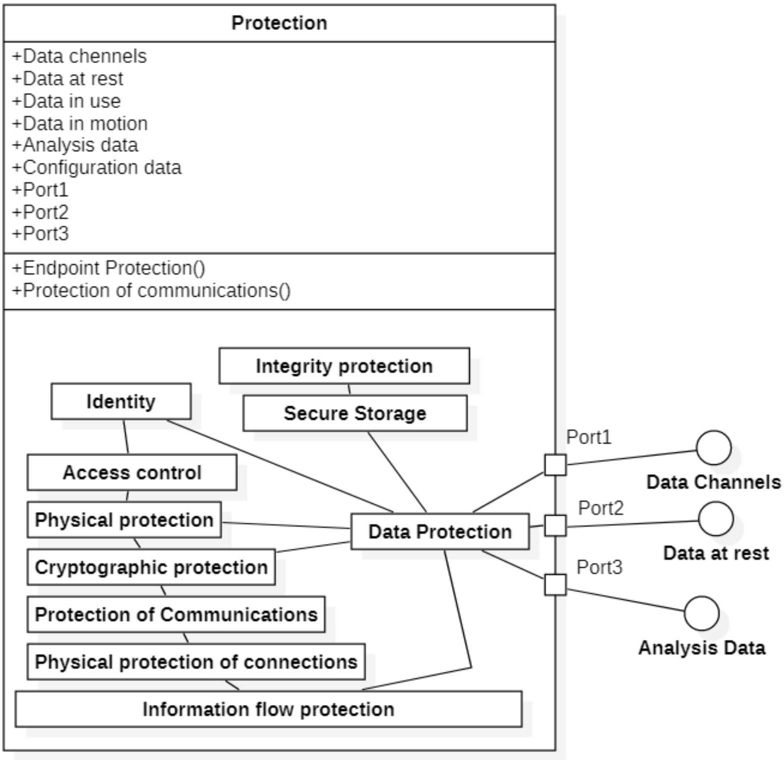

Figure 6: UML Composite structure diagram” of the class “Protection”.

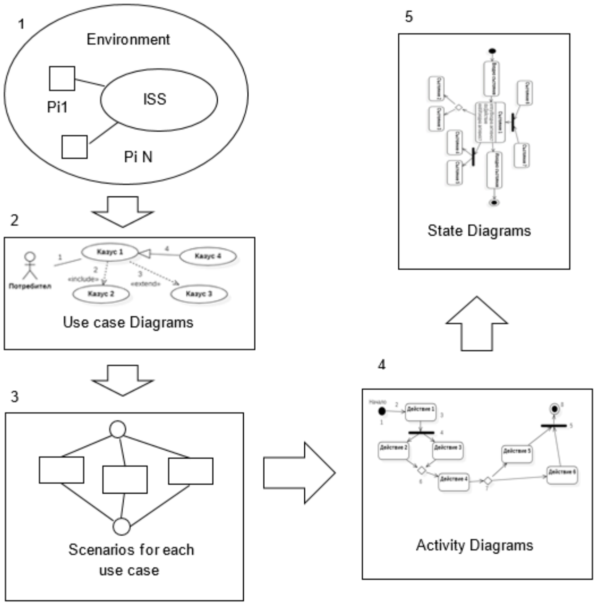

Figure 7: Approach to creating a functional model using dynamic UML diagrams.

Agent-based Simulation Model

The agent approach is based on "agents" – any set that perceives the environment through sensors, processes information, and influences the environment through actuators (Fig. 8). The agent-based modeling approach is used to create simulation models to pre-simulate the operation of the designed ISS.

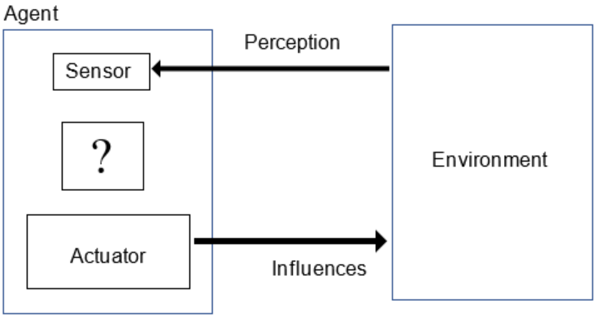

Figure 8: Agent.

In the process of ISS design, we assume that certain separate agents performing independent tasks to protect a given asset or information are part of the system. In order to achieve the objectives of the ISS, the system must ensure

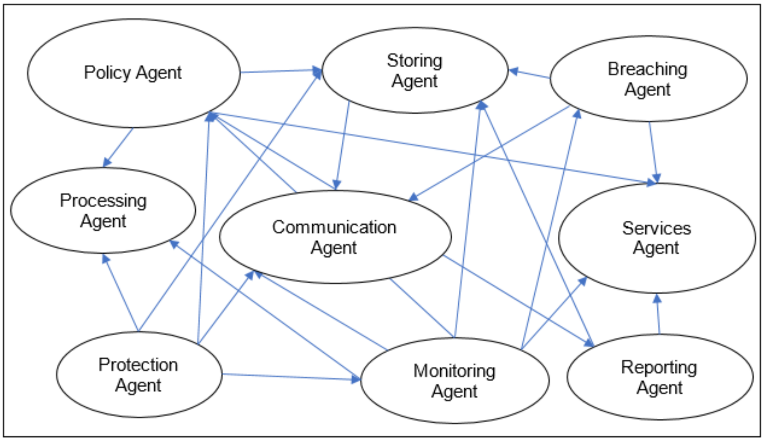

Figure 9: Relationships in the "Agent" class.

the activity of several types of agents performing a certain role and some interactions between them. The following agents can be defined: "Policy Agent," "Storing Agent," "Breaching Agent," "Processing Agent," "Communication Agent," "Services Agent," "Protection Agent," "Monitoring Agent," and "Reporting Agent", which are forming class "Agent" (Fig.5). The relationships of the agents are shown in Fig.9.

Transformation of the Design Model into an Agent-based Simulation Model

The transformation of the object-oriented project model of the ISS into an agent-based simulation model is based on the class-diagram of Fig.5. It is evident that the class "Control" contains objects that have the potential to become pro-active agents, making independent decisions. This class interacts with the other classes (Security, Data, and Management), which shape the environment of the agent class and provide them with the information they need to work and make decisions. In addition, the agents act on the environment in order to achieve specific results for ISS and receive information for the decisions. It can be assumed that these classes form the environment with which the Control class interacts. So, a class "Environment" can be defined as consisting of three classes – Protection, Data, and Management. The Control class then becomes a Control Agent, interacting with the new Environment class (Fig.8). On this basis, it can be assumed that the "Control" class is transformed into an "Agent" class, which represents a set of agents, each with a specific purpose. In accordance with the individual objectives and the need for certain roles, the class "Agent" includes: "Breaching Agent," "Protection Agent," "Policy Agent," "Monitoring agent," "Reporting agent," "Communication Agent," "Processing agent," "Storing agent," and "Services agent."

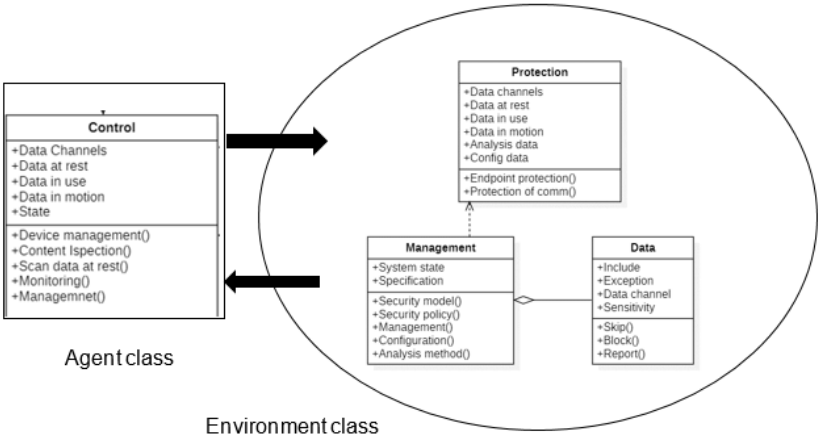

Figure 10: "Agent" and "Environment" classes.

To describe the role of each agent by using the UML language, a use case diagram is used, which describes the interaction of an agent with the environment and in our case is considered as an OO system. For each of the listed agents, a diagram is created that describes the scenario in which it operates. To show how this is done, the 'use case' diagram is presented to the environment by the 'Policy Agent' and 'Protection Agent' agents. The other agents are described in a similar way, which creates an agent-based model of ISS implementation. This model is used to simulate the activity of the designed system.

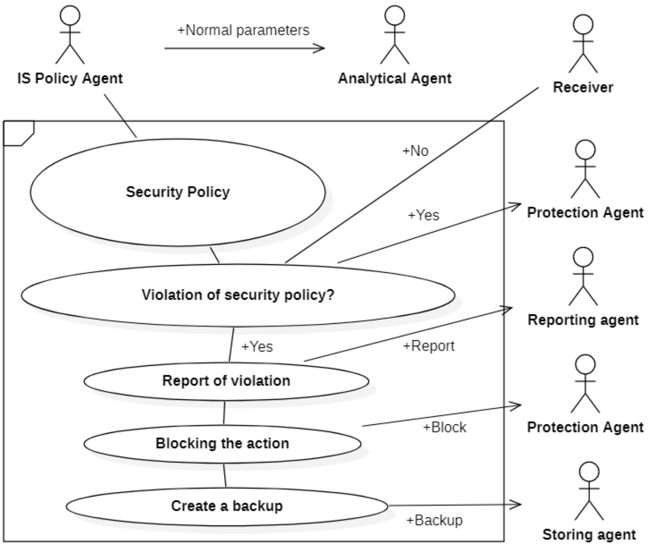

Figure 11: Use case diagram for Policy Agent.

Fig. 11 shows a use case diagram for "Policy Agent." The scenario shown by the diagram is as follows: "Policy Agent" monitors security policy compliance. In the event of a violation, the "Protection Agent" is activated and, according to the actions provided in the security policy, activates the following agents: "Reporting Agent," to prepare a report on the violation, "Protection Agent" to block the action, "Storage Agent" to store the report and a copy of the document that violated the security policy for further investigation. The agent periodically submits the parameters indicating the normal operation of the system to the Violation Agent, serving to detect a deviation and identify a violation accordingly.

Fig. 12 shows a use case diagram for "Protection Agent." The agent is activated by the Violation Agent upon detection of a deviation from normal parameters. The agent initiates the implementation of the previously planned actions to counteract the violation or threat.

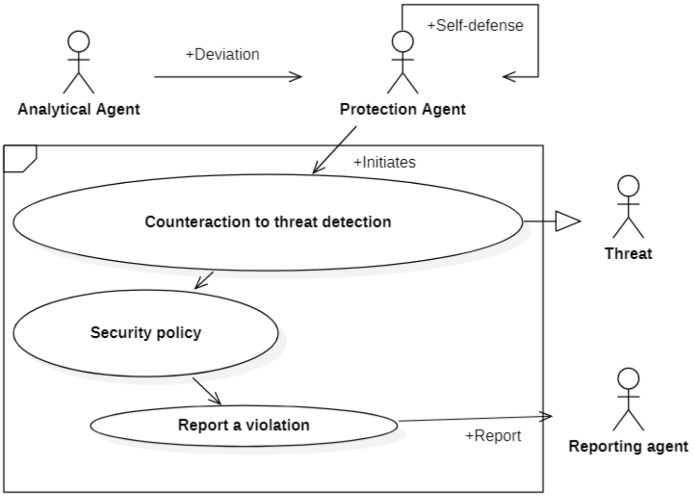

Figure 12: Use case diagram for Protection Agent.

Conclusions

Why is the transformation of a UML design model into a simulation model necessary? When designing a modern complex ISS, it is necessary to take into account the various requirements of the various stakeholders. It is very important to be sure that the designed system will satisfy their requirements, as well as a number of external requirements – technological and normative. Therefore, a preliminary simulation of the system's operation is important to make sure that these requirements will be met. The proposed transformation is part of our proposed comprehensive method for designing an information security system in organizations. Our method has good practical application in the fields of continuous improvement of cybersecurity processes, complex systems analysis, design and development, risk analysis, and risk assessment.

The approach is applied in system simulation through various agent-based simulation environments.

References

[1] Accenture Security, "2019 Cyber Threatscape Report," https://www.accenture.com/_acnmedia/pdf-107/accenture-security-cyber.pdf.

[2] Vladimir Shangin, *Zastita Informacii v komputernyih sistemah i setyah* (DMK Press, 2012).

[3] Mark Maier, David Emery, and Rich Hilliard, "ANSI/IEEE 1471 and Systems Engineering*," Systems Engineering* 7, no. 3 (2004): 257-270, https://doi.org/10.1002/sys.20008.

[4] "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," IEEE 1471-2000, 2000, https://standards.ieee.org/standard/1471-2000.html.

[5] "ISO/IEC/IEEE 42010:2011 – Systems and Software Engineering – Architecture Description," 2011, https://www.iso.org/standard/50508.html.

[6] The Unified Modeling Language (UML) Web Page, accessed April 11, 2022, https://www.uml-diagrams.org.

[7] Alan Dennis, Barbara Haley Wixom, and David Paul Tegarden, *System Analysis & Design – An Object-Oriented Approach with UML*, 5th Edition (John Wiley & Sons, 2015), 19-52.

[8] Ivan Gaydarski, Zlatogor Minchev, and Rumen Andreev, "Model Driven Architectural Design of Information Security System," *Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR)* 2018, pp. 349-359, Springer, 2019, https://doi.org/10.1007/978-3-030-17065-3_35.

[9] Ivan Gaydarski, Zlatogor Minchev, and Rumen Andreev, "Model Driven Approach for Designing of Information Security System," *Journal of Information Assurance and Security* 13 (2019).

[10] "Guidelines on assessing DSP and OES compliance to the NISD security Requirements," ENISA, November 2018, https://www.enisa.europa.eu/publications/guidelines-on-assessing-dsp-security-and-oes-compliance-with-the-nisd-security-requirements.

[11] "Standards, Guidelines, Tools and Techniques," ISACA, May 2016, www.isaca.org/resources/isaca-journal/issues/2020/volume-6/standards-guidelines-tools-and-techniques.

About the Authors

In 2022, **Ivan Gaydarski** received a Ph.D. degree from the Institute of ICT, Bulgarian Academy of Sciences. He is a co-founder and managing director of Infinity Ltd. – a company specializing in the implementation and maintenance of IT security solutions with a focus on data security, monitoring and analysis of information flows, risk assessment, and security policymaking. Infinity is VAR of world-leading IT Security vendors. It consults, delivers, implements, trains, and supports organizations and structures from different sectors and industries in Bulgaria and abroad, including governmental, military, national security, aviation, healthcare, finance, insurance, education, and media organizations.

Pavlin Kutinchev is a Ph.D. student at the Institute of ICT, Bulgarian Academy of Sciences. He is also a co-founder and managing director of Infinity Ltd.
E-mail: pavlin.kutinchev@iict.bas.bg