MODEL - AND SIMULATION DRIVEN SYSTEMS - ENGINEERING FOR CRITICAL INFRASTRUCTURE SECURITY ARCHITECTURES

Sascha Goldner and Philipp Rech
System Design Center, EADS Deutschland GmbH, Landshuterstraße 26, Unterschleißheim, Germany

Keywords: Critical Infrastructure Protection, Scenario based, Rule based, Process Modelling, Case based Reasoning, Expert System, Systems Engineering, NATO Architecture Framework, CONOPS, Simulation.

Abstract: The design of systems that support the protection of critical infrastructures or state borders against various threats is a complex challenge. This requests the use of both modelling and simulation in order to reach the operational goal. Threat scenarios like terrorist attacks, (organized-) crimes or natural disasters involve many different organisations, authorities, and technologies. These systems are often operated only by implicit known processes and diverse operational guidelines based on dissimilar legislations and overlapping responsibilities. In order to cope with these complex infrastructure systems and their interconnected processes, a scenario- and architecture based systems engineering approach must be implemented to design a solution architecture compliant with the requirements, internal and external demands. This paper gives an overview of the developed approach towards the system architecture and explains the different engineering steps in order to implement this architecture for real use-cases.

1 INTRODUCTION

Maintaining and improving the security of critical infrastructures against various known and new threats is a crucial task and often just tackled by the implementation of new technology without assessing the effectiveness of the security measures as a whole. Besides security technologies (e.g. CBRNE scanners, CCTV, etc.), nowadays rules and regulations also have to be considered in the implementation of security systems and all related processes in critical infrastructures. Moreover, the many different involved authorities (i.e. airport operators, local and federal police, different fire departments, military etc.), work by experience and implicit knowledge with regards to their organisation-specific guidelines for decision making. Due to the great importance of these non-technical factors of influence, there is a need for a security system engineering approach that takes into account these non-technical parameters and their harmonization among the different organizations participating.

Within this paper we present the EADS engineering approach based on the usage of an expert system, an architecture framework as well as a simulation tool and describe in detail the different engineering steps. Currently this approach is successfully implemented within studies and international large-scale projects in the domain of border security and critical infrastructure security.

2 OBJECTIVE

Our objective is the development of system architectures for complex security systems. This requires the detailed modelling of the respective security mechanisms and relevant processes in which the many different authorities are involved. Their expert knowledge, the real-world behaviour and all required decisions have to be gathered, analyzed and represented in a consistent way. By application of our method the following goals will be met:

- Gathering tacit expert knowledge and transforming it into system / software behaviour
- Improving of the functionality of security systems against new emerging threats by using

Goldner S. and Rech P. (2010).
MODEL - AND SIMULATION DRIVEN SYSTEMS - ENGINEERING FOR CRITICAL INFRASTRUCTURE SECURITY ARCHITECTURES.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 424-427
DOI: 10.5220/0002972604240427

new technologies (e.g. 'naked' scanner) and processes.
- Using of domain expert knowledge in order to verify and validate the system architecture

3 APPROACH

In order to reach the above described objectives we developed a generic engineering approach (Fig. 1) which is implemented in six subsequent steps using an EADS-tailored tool chain. The following paragraphs will explain the steps in detail.

Step 1: Analysis of As-is System. A fundamental precondition of all engineering approaches is a deep knowledge about the system or object of investigation. For this reason the first step of the engineering approach is a system analysis in order to understand the system itself and its functioning. In a data gathering phase of this analysis all relevant elements and parameters are identified which describe and influence the object of investigation (e.g. airport, border control, etc.). Further more, it is analysed how these parameters correlate and influence each other. The direction of a correlation is defined and also a qualitative assessment is applied which allows to describe the degree of correlation of the parameters (no, medium, strong correlation). The influence analysis can be done by using either correlation tables or matrices.

Another aspect of the system analysis concentrates on the business processes. They describe the activities and their sequence of execution in order to produce a specific service or product on a very fine-grained level. For every process step it has to be determined e.g. who performs this step, who is responsible, what are the input and output parameters, how long does this step take and what systems and tools are used to implement this process activity.

The step of analysing the target system is crucial and the analysis results define the basis for creating threat scenarios which is done in step 2.

Step 2: Scenario Generation. In our work we pursue a scenario-driven approach because we believe that only a real-life scenario can show the actual benefit and practical relevance of the method. Further we need a real-life reference in order to demonstrate how the knowledge of domain experts is incorporated in our work. Finally the added value of our method can be measured and compared more easily when it is based on a real-life example and it is more comprehensible to any stakeholder.

A scenario is used to describe a potential threat respectively a potential attack that can be carried out against the security system. The scenario description is made up of the system elements that have been specified in the previous step. The important point is the ability to create self-consistent scenarios. This is because of the correlations between the system elements that have been specified in step 1. For example, it makes no sense to use a knife in order to attack an armoured vehicle. In that case, we would have defined a negative correlation value between knife and armoured vehicle.

Another very important aspect of the scenario-driven approach is that the business processes which are affected by the scenario can be derived automatically. Regarding business processes, we act on the assumption that there are just a few roles that can be taken on in the system, e.g. the system "airport" offers for example the roles of a passenger, an employee, a visitor and a supplier. Further we assume that an attacker tries to remain undetected until the actual attack on the target starts by slipping into at least one of the roles the system provides. Just in the moment the attacker performs the attack or the attacker's cover has been compromised, the attacker's original role becomes obvious.

Business processes describe the intended behaviour of a system and our goal is to create a system architecture which on the one hand provides a functionality that complies with the real-life behaviour and on the other hand allows to assess and optimize the systems security mechanisms. Our main conclusion in this regard is that the system architecture has to have a high degree of structural similarity to the systems business processes. The next section describes how such a system architecture can be achieved. The result of this step are self-consistent scenarios.

Step3: Design of the Solution Architecture. After the as-is system is analysed and threat scenarios are defined, the goal of the third step in the system engineering approach is the design of the desired to-be system architecture. In this architecture we consider two different views (Figure 1):

- Operational View
- System View

The operational view describes how the roles, organizations and activities have to operate in order to react on the given threat. This description provide the operational guidelines which are noted in a CONOPS (concept of operations). The system view specifies the technical framework for the implementation of the CONOPS. This system view

consists of different components such as platforms (e.g. aircraft-carrier), hardware (camera, radar), software, facilities and material.

Both the operational and the system view of the architecture have to be designed in a way that they enable an organisation to cope with the scenarios defined in step 2.

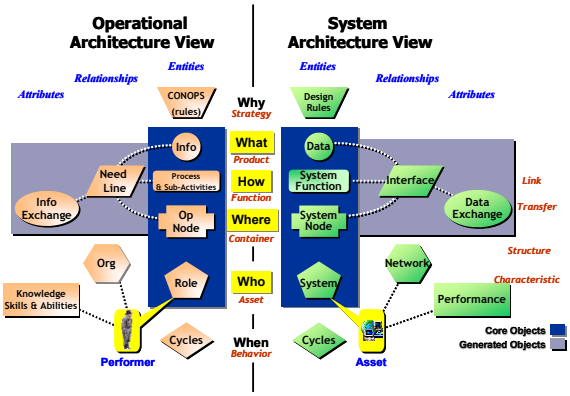

Figure 1: Elements within the Operational- and System Architecture (Source: NATO Architecture Framework v3).

First the operational view has to be defined in order to support the realization of the desired CONOPS. Therefore the required capabilities of an organisation have to be gathered. These are for example surveillance and reconnaissance in the military or evacuation and crisis management in the civil environment. In the system engineering methodology these capabilities will be modelled using a capability taxonomy.

In a next step the operational nodes have to be defined which are needed to realize the required capabilities. Operational nodes are processes or rather a set of successive activities with information flows between them in order to exchange data. This analysis provides a clear picture how operations are performed and thereby support the system engineer to specify the required technical equipment to carry out the processes.

The result of step 3 is (amongst other architecture elements) a process model that shows all units of behaviour (actors, systems, processes) and their condition-based interactions.

The more challenging task is the acquisition of the detailed behaviour for the decision nodes in that process model. These are required in order to perform a realistic simulation and optimization of the process. To cope with this challenge EADS has build an interface between the architecture modelling tool and a case-based expert system which is used to enter the business logic / behaviour for the decision nodes of the process model. This procedure is described in more detail in the following section.

Step 4: Verification and Validation of the System Architecture. After the process model has been created, the correctness of the architecture has to be verified in order to provide the functionality the domain experts expect. The step of verification and validation is a quite straight forward task when dealing with pure technical systems. In domains where implicit knowledge and human decision making is part of the overall system behaviour, a formal verification and validation is difficult to apply. Far more feasible is the approach of using concrete examples to validate the system behaviour. The test data are created by the domain experts, thereby ensuring that their implicit knowledge is incorporated into the system functionality. In our approach we use an expert system to validate and verify the system architecture. This modus operandi is actually a two-step process. In the first step a behaviour model is created within the expert system whose structure conforms to the system architecture. In the second step the functionality or behaviour of this architecture is implemented, verified and validated.

In order to create a behaviour model within the expert system the process model of Step 3 is imported and displayed as a graph. The behaviour of each model element is created by defining rules for specific situations (case-based reasoning). Based on these rules, decisions are made how to process a specific input and where to send the result. The model structure remains the same, but depending on the input data respectively the situation, the output might be different. The two main features of this method are as follows:

- Data-driven behaviour modelling: Based on our experiences we realized that knowledge acquisition and its implementation into systems can be achieved more easily by using real-life examples. Examples are close to reality and easier to describe and to verify compared to general and highly theoretical statements. Therefore we use a set of training data, so called training situations, in order to create a node's desired behaviour and thereby to train the overall model.
- Consistent behaviour modelling: while defining a rule for a process node the rule engine automatically checks the consistency of that rule with respect to all the existing rules. If an inconsistency is detected the user is forced to change the rule until the inconsistency is eliminated. This mechanism is called

"conclusion and justification". The user has not only to define what decision based on the input data has to be made, but moreover he has to define why it has to be made. To justify a conclusion the specific training situation has to be used and therefore it has to be sufficient.

Once the behaviour model has been created, the expert is able to validate and verify the behaviour by using the test data in order to compare the results of the behaviour model with the intended system. A further output of step 4 are executable rules which are used as parameters for the simulation in the next step.

Step 5: Simulation of the Process- and System Configuration. After the rules have been defined using the case-based reasoning approach the process diagrams have been transferred into a simulation framework. This simulation is used to analyse and optimise different process and technology configuration alternatives based on time, resources and other (changeable) parameters.

Using process simulation we are able to determine weaknesses in some scenarios and suggest and validate an optimized alternative process configuration. Further an optimal resource allocation can be computed.

These results give important hints and justifications to the desired target architecture as described in the next section.

Step 6: Refinement of the Target Architecture. Using the described engineering steps we are able to define an architecture that is consistent and fulfils the operational and system requirements. The consistency is reached by carefully designing the architecture using an architecture model and modelling tool environment that provides consistency checks and gap analysis reports. The use of simulation enables us to define an optimal architecture configuration and resource assignment.

The use of case-based reasoning enables us to create and manifest a consistent business logic that is very close to the way the human expert works because it is gathered in a machine learning approach in which all justifications are visible and traceable.

All previous steps (1-5) are performed iteratively and enforce a constant update and refinement of the requirements and the system architecture similar to an agile method well known in software design.

4 CONCLUSIONS

Constantly new emerging threats in the domain of infrastructure and border security require existing and future security systems to be able to cope with those new threats. Our engineering approach accommodates these new requirements and enables the engineering of a highly functional security architecture by:

- Applying case-based reasoning in order to gather human behaviour and tacit knowledge from domain experts.
- Providing iterative and agile engineering steps based on modelling and simulation.
- Using military and civilian modelling frameworks and standards for the architecture models.
- Validation and verification of the system functionality on real case scenarios and expert knowledge.

REFERENCES

Cole, M., et al., 2004. Optimisation of Critical Infrastructure Protection: The SiVe Project on Airport Security. In *CRITIS'09, 4th International Workshop on critical information infrastructure security.* Springer.

Wisnosky, D., 2004. *DoDAF Wizdom: A Practical Guide,* Wizdom Press.

International Council on Systems Engineering (INCOSE), 2010. *INCOSE Systems Engineering Handbook v3.2*, INCOSE.

NATO Consultation, Command and Control Board under cover of AC/322-D(2007)0048-AS1, 2007. *NATO Architecture Framework Version 3*.