An Educational Framework to Support Industrial Control System Security Engineering

Thesis submitted for the degree of
Doctoral of Philosophy at De Montfort University

By

Nuria Mahmud Benjuma

Department of Software Technology Research Laboratory (STRL)
Cyber Technology Institute (CTI)
Faculty of Technology
De Montfort University-Leicester
United Kingdom

JUNE 2017

Abstract

Industrial Control Systems (ICSs) are used to monitor and control critical infrastructure such as electricity and water. ICS were originally stand-alone systems, but are now widely being connected to corporate national IT networks, making remote monitoring and more timely control possible. While this connectivity has brought multiple benefits to ICS, such as cost reductions and an increase in redundancy and flexibility, ICS were not designed for open connectivity and therefore are more prone to security threats, creating a greater requirement for adequate security engineering approaches.

The culture gap between developers and security experts is one of the main challenges of ICS security engineering. Control system developers play an important role in building secure systems; however, they lack security training and support throughout the development process. Security training, which is an essential activity in the defence-in-depth strategy for ICS security, has been addressed, but has not been given sufficient attention in academia. Security support is a key means by which to tackle this challenge via assisting developers in ICS security by design.

This thesis proposes a novel framework, the Industrial Control System Security Engineering Support (ICS-SES), which aims to help developers in designing secure control systems by enabling them to reuse secure design patterns and improve their security knowledge. ICS-SES adapts pattern-based approach to guide developers in security engineering, and an automated planning technique to provide adaptive on-the-job security training tailored to personal needs.

The usability of ICS-SES has been evaluated using an empirical study in terms of its effectiveness in assisting the design of secure control systems and improving developers' security knowledge. The results show that ICS-SES can efficiently help control system designers to mitigate security vulnerabilities and improve their security knowledge,

reducing the difficulties associated with the security engineering process, and the results have been found to be statically significant.

In summary, ICS-SES provides a unified method of supporting an ICS security by design approach. It fosters a development environment where engineers can improve their security knowledge while working in a control system production line.

Acknowledgement

I would like to express my sincere gratitude to my supervisors Dr. Richard Smith and Prof. Helge Janicke for the continuous support throughout my PhD study, for their patience, motivation, and immense knowledge.

I gratefully acknowledge my sponsor: the Libyan embassy in London for funding my PhD. Many thanks to Almergib Univerisity and Higher Education Ministry for the PhD Scholarship.

A special thanks and appreciation to my beloved husband Dr. Abdlrzag Ehdode, my lovely son Ahmed and beautiful daughter Abrar for endless support and all of the sacrifices that they have made during this journey.

Words cannot express how grateful I am to my beloved parents, my brothers, and sisters for the numerous support and encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

This chapter introduces the research problem and formulates the thesis statement and research questions. The chapter is divided into the following sections: Section 1.1 gives the motivation for this research. Section 1.2 discusses the research problem. Section 1.3 formulates the thesis hypothesis. Section 1.4 defines the research questions. Section 1.5 presents the objectives. Section 1.6 defines the success criteria. Section 1.7 highlights the thesis contributions to the current literature. Section 1.8 outlines the structure of this thesis, providing an overview for each chapter.

1.1. Motivation

Industrial Control Systems (ICS) are used for controlling critical infrastructure such as water and waste water, electricity, oil and natural gas (Stouffer et al., 2011). Initially, ICS were isolated systems in physically secure areas (Stouffer et al., 2011). Since these systems are now being widely connected to IT networks so as to use web applications and services to remotely monitor and control ICS data, the possibility of ICS security vulnerabilities and incidents have been significantly increased, creating a greater need to secure and adequately protect these systems (Stouffer et al., 2011).

Recent incidents such as the Stuxnet attack, which disrupted a uranium fuel enrichment plant in Iran (Creators, 2013), and the Slammer worm, which disabled a nuclear power plant in Ohio (Collins and McCombie, 2012), have shown that control systems are vulnerable when not sufficiently secured. ICS attacks can cause serious effects to the economy and even to human lives (Stouffer et al., 2011). These attacks show the limitations of current ICS security engineering and vulnerability detection (Kargl et al., 2014). A rigorous research on security approaches and technologies is required in response to the dramatic increase in the number of cyber security threats to critical systems (Abouzakhar, 2013).

There are several factors that can play important roles in securing control systems (Stouffer et al., 2011). Applying security throughout the system development cycle can reduce the possibility of producing security vulnerabilities (Lemaire et al., 2014). Developers' security knowledge is also particularly important in reducing security

weaknesses in control systems, as recommended by the defence-in-depth strategy (Stouffer et al., 2011). Knowledge management and training competencies were identified as key requirements and features in improving ICS security (Hentea, 2008) (Steven, 2006).

In this thesis, a novel supported framework, Industrial Control System Security Engineering Support (ICS-SES), was proposed with the intention of providing a mechanism to support a control system security by design approach. The framework is based on security patterns, which capture security expertise in the form of reusable solutions and an automated planning technique for providing tailored training. These techniques, if adopted, can provide effective help in designing secure control systems.

1.2. Problem Statement

The problem being addressed in this research is that industrial control systems lack security engineering. Although developing secure ICS has been the interest of many researchers in both industry and academia (Drias et al., 2015) (Axelrod, 2011) (Kunsman et al., 2015), there is a knowledge gap within control system security engineering research, and more effort is needed, in particular, in the area of ICS security by design (Hadziosmanovic et al., 2012).

While there are some tools that have been developed to support ICS security, such as CSET in reference (ICS-CERT) and AVATAR in reference (Pedroza et al., 2011) for SYSML designers, current tools do not focus on the security awareness and learning aspect (Foo et al., 2013).

The culture gap between ICS developers and security experts can be bridged by a pattern-based security engineering approach (Stouffer et al., 2011). However, current research lacks practical guidance on selecting and applying security patterns into control system development processes (Nguyen et al., 2014).

Therefore, this research was carried out using a design science research methodology, with the aim of helping to fill the knowledge gap in ICS security engineering research by proposing a novel supported method for control system security engineering.

1.3. Research Hypothesis

Technology can be used to support developers in the design of secure control systems and in improving their security knowledge. The resulting supported framework was defined as the Industrial Control Systems Security Engineering Support, 'ICS-SES'. The argument is that ICS-SES can assist engineers in developing secure control systems. ICS-SES is usable and can effectively help developers improve their security knowledge and design secure control systems.

1.4. Research Question

The following research questions have been formulated to support the thesis statement:

1. What is the state-of-the-art in control system security engineering?
2. What are developers' needs regarding the design of secure control systems?
3. Can an on-the-job adaptive tool be created to support control system security by design?
4. Can a supported tool assist developers in designing secure control systems?

1.5. Research Aim and Objectives

The aim of this research is to develop a solution to support and improve control system security engineering as a response to the shortcomings in the current development approaches.

This aim is supported by the following objectives:

1. To systematically review the literature concerning ICS security engineering, ICS security by design, challenges in developing secure ICS, using security patterns in ICS development, and ICS security support and training.
2. To understand the current level of developers' security awareness and knowledge.
3. To understand control system developers' needs in designing secure control systems.
4. To explore the existing support methods for control system security engineering.

5. To provide an on-the-job technical support mechanism for designing secure control systems.
6. To provide on-the-job adaptive security training tailored to personal developers' needs.

1.6. Success Criteria

This section indicates the criteria that will be used to measure the success of this research. These criteria are formulated as follows:

- Guiding control system developers in selecting secure design patterns.
- Providing security training material in understandable language for control engineers.
- Providing training material related to the problem and design context.
- Providing training material tailored to users' personal needs.
- Running with acceptable system performance.
- Ease of use of the supported tool.
- User satisfaction.

The success criteria were measured by evaluating the proposed method through an experimental evaluation study.

1.7. Contribution of the thesis

This section outlines the contributions of this thesis as the following:

- A novel adaptive tool that supports control system security engineering throughout the development cycle.
- A novel on-the-job guide using secure design patterns in control system development.
- The use of on-the-job security training in the discipline of control engineering.
- A new combination of security pattern guide and embedded training aid.
- Contextualised and personalised security training for industrial control systems.
- Bridging the gap between security experts and control system engineers.
- Improving comprehension of control system security.

1.8. Thesis Structure

This section briefly summarises the thesis structure. This thesis is organised into eight chapters in line with the research objectives.

Chapter Two: Systematic Literature Review

This chapter presents a systematic literature review concerning control system security engineering issues and challenges. The chapter provides essential background information. It identifies the state-of-the-art in ICS security by design and highlights the limitations of previous work on ICS security engineering in the literature.

Chapter Three: Research Methodology

This chapter discusses information system research paradigm candidates and justifies the selection of the Design Science Research (DSR) methodology to conduct this research. It introduces our research design in relation to the selected methodology.

Chapter Four: A qualitative study of Control System Developers' Support Needs for Security Engineering

This chapter introduces a qualitative study designed to enrich the understanding of ICS security engineering and identify current levels of security awareness and knowledge amongst control engineers. It identifies developers' needs regarding the design of secure control systems.

Chapter Five: Framework

The chapter introduces the proposed Industrial Control System Security Engineering Support (ICS-SES) framework. It demonstrates the proposed pattern-based security guide and tailored training method. The chapter explains the workflow required to support the ICS security engineering process.

Chapter Six: A Controlled Experiment for Evaluating the ICS-SES framework

This chapter presents an empirical study to evaluate the usability of the proposed framework. The chapter discusses the experimental design and procedure and illustrates its execution.

Chapter Seven: Analysis and Evaluation

This chapter analyses and discusses the results. It evaluates the ICS-SES framework and discusses the research findings.

Chapter Eight: Conclusion and Future Work

This chapter concludes this thesis and discusses the research contributions and directions for future work.

Chapter 2
Literature Review

Chapter Objectives

- To provide essential background information.
- To identify the body of work related to the research questions
- To identify the state-of-the-art of ICS security by design
- To identify the gap in the knowledge and limitations of previous work.

2.1- Introduction

Industrial Control Systems are used for monitoring and controlling critical infrastructures that provides nations with essential resources such as electricity and water. If these systems stop working properly, the consequences could be disasterous: significant equipment damage, serious environmental damage or even death.

In the past, Industrial Control Systems, 'ICS', were initially built as standalone systems and were not connected to the internet, which made security considerations unimportant in control systems development (Drias et al., 2015). As these systems are now being incorporated into Wide Area Networks (Orlikowski and Baroudi) and are thus potentially reachable by malicious internet users, ICSs are becoming increasingly at risk from cyber-attack (Durrani et al., 2013).

This chapter introduces a systematic literature review on the area of ICS security research. The review was conducted based on the guidelines identified by Keele (Keele, 2007) and Kitchenham *et al* (Kitchenham et al., 2009). The methodology of the review is discussed in Chapter 3, Section 3.4.

2.2. Background

An Industrial Control System (ICS), consists of several types of field devices that are supervised from a centralised location (Galloway and Hancke, 2013) (Fovino et al., 2010). ICS are typically used for remotely monitoring and controlling critical

infrastructure such as water and wastewater treatment, chemical, oil and natural gas, transportation, power stations and discrete manufacturing. ICSs encompass several types of control systems, such as distributed control systems (DCS), and Supervisory Control And Data Acquisition (SCADA) systems as shown in Figure 2.1. The main distinction between these two systems is that SCADA systems are more geographically distributed than DCS (Stouffer et al., 2011) (Krotofil and Gollmann, 2013).

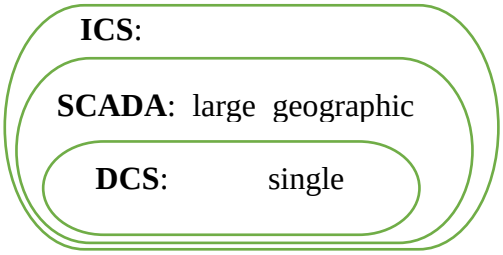


Figure 2-1 Industrial Control Systems (Stouffer et al., 2011)

The following subsections explain the components of ICS and address the differences between ICS and Information Technology (IT).

2.2.1- Common Industrial control Systems components

This section presents ICS Architecture, which includes control components and industrial network components.

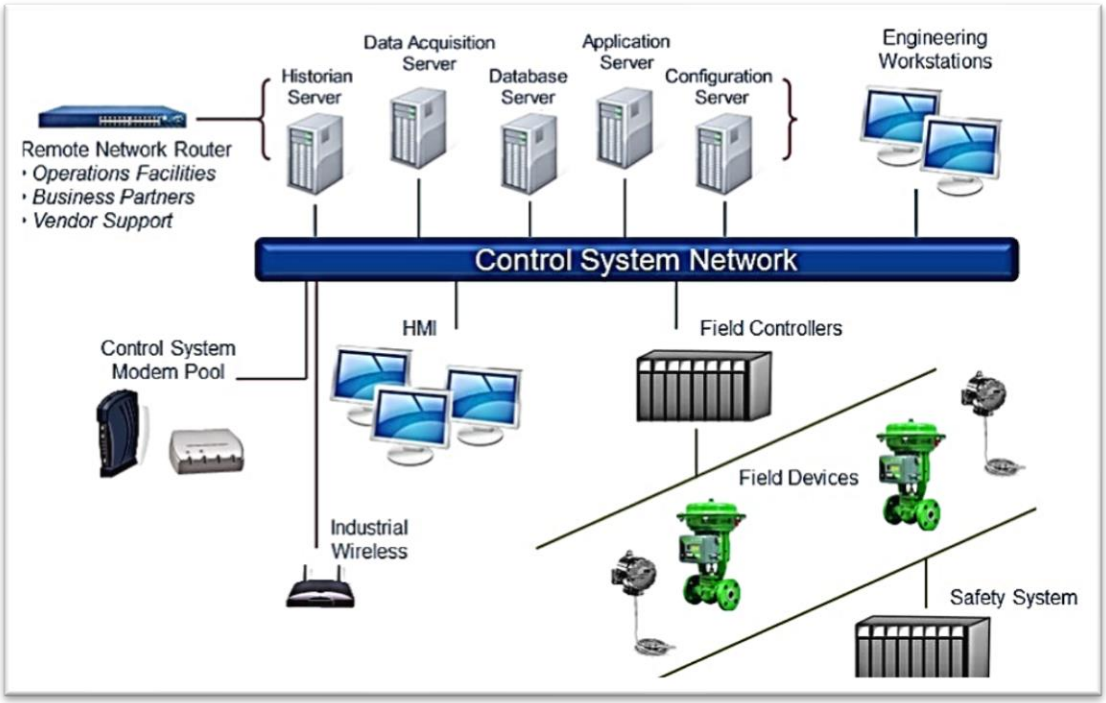


Figure 2-2 ICS Architecture (ICS-CERT)

ICS components are used to control and monitor field devices, as shown in Figure 2.2 (ICS-CERT).

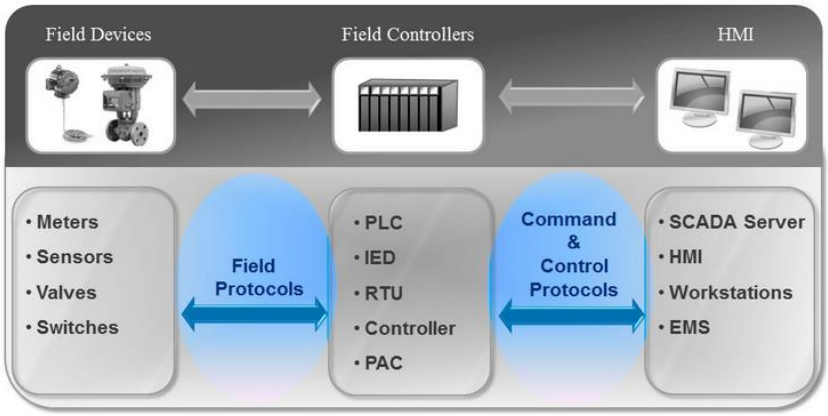


Figure 2-3 ICS Process (ICS-CERT)

Figure 2.3 shows that ICS architecture consists of three main components:

Field Devices: are the interface between physical processes and control systems. They include input devices such as sensors and measuring instruments that measure the device outputs which control process parameters and actuators, as shown in figure 2-4.

Figure 2-4 Input and Output Devices in ICS (ICS-CERT)

Field Controllers: these components control the communication between field devices and Human Machine Interface (HMI). They collect input and output data from field devices and send it to the HMI that accordingly issues process control commands and send these to the field controllers. In large distributed systems, the field controllers may collect and process information from hundreds of field devices. They are often located close to field devices in order to be able to perform rapid communications.

Field Controllers are embedded microprocessor devices. They convert electrical signal ‘input data’ received from field devices into digital signals and convert the digital signals received from HMI into ‘output data’. The four main types of field controllers are: Programmable Logic Controllers (PLC), Intelligent Electronic Devices (IED), Programmable Automation Controllers (PAC), and Remote Terminal Units (RTU).

Human Machine Interface ‘HMI’: is a user interface that provides a graphical visualisation of industrial monitoring and control systems. HMIs allow operators to view real-time or near real-time process information. HMIs are typically run from computers from such devices as touch panels or software-based applications on personal computers, smartphones, tablets, or workstations. HMIs are used by operators to control and monitor processes through their communication with field controllers such as PLC, PAC and RTU. They are capable of supporting other applications and providing historical trends, event notifications and alarms.

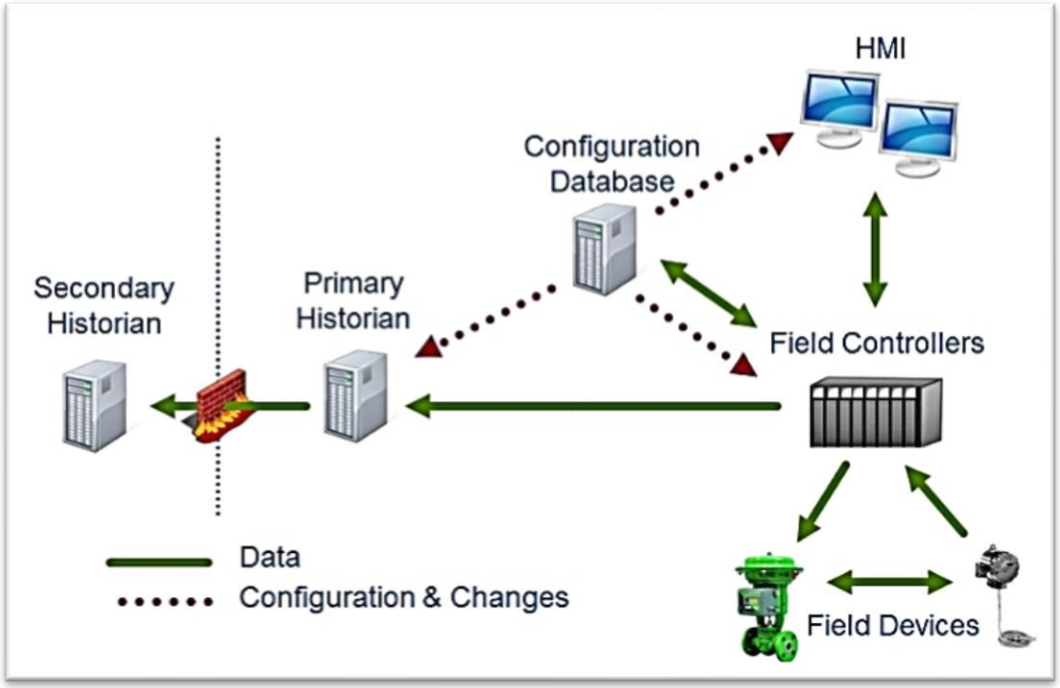


Figure 2-5 Data Flow within an Industrial Control System (ICS-CERT)

Figure 2.5. articulates an example of the communication between ICS components. The field devices send process data to field controllers. Field controllers transmit these data and send them to the related component(s). They send real-time process data to the HMI, historical process data to the historian and hardware error statuses to the configuration database (ICS-CERT).

From a security perspective, HMI systems and data are obvious targets for cyber-attacks, as they are usually connected to outside networks or are accessible via remote access methods. This interconnectivity could allow an attacker to take over critical system processes (ICS-CERT).

2.2.2- Industrial Control Systems and Information Technology.

Industrial Control Systems adopt IT solutions to increase their capability in terms of remote access and corporate connectivity. They are being developed using Operating Systems (OS), industry standard computers, and network protocols (Stouffer et al., 2011). This integration promotes the associated IT capabilities; however, it also makes ICSs less isolated from the outside world, increasing the need to secure these systems (Stouffer et al., 2011).

Figure 2.6 shows the most common elements in IT architecture. It helps to identify the similarities between IT and ICS architectures as this is paramount to addressing cybersecurity strategies within control systems. The diagram articulates how system components communicate within both the business world and control systems (ICS-CERT).

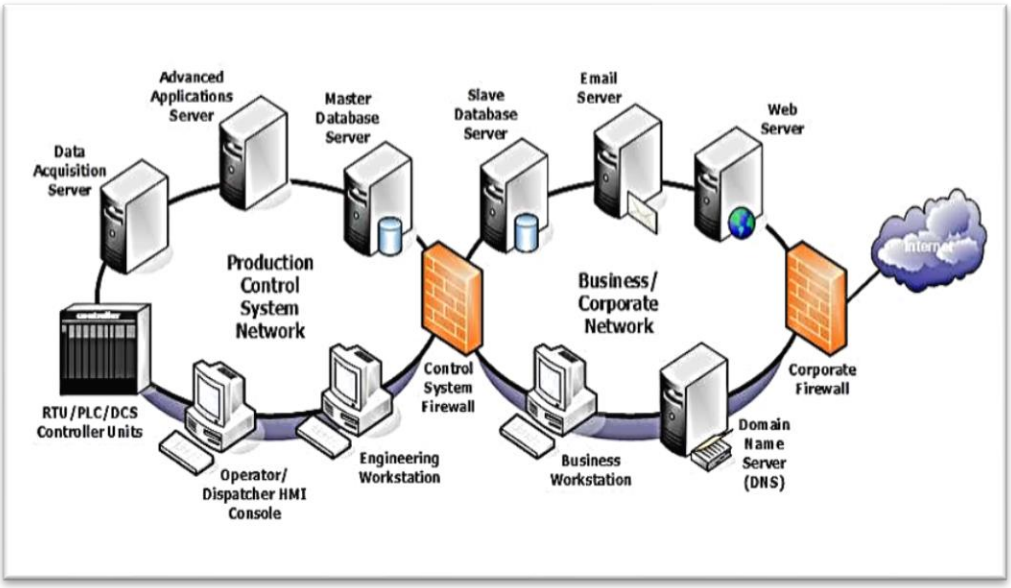


Figure 2-6 IT architecture and control system architecture (ICS-CERT).

Despite a number of similarities between these two systems, Industrial Control Systems similar differ from standard IT systems in many ways, such as priorities and risks. Control systems have different reliabilities and performance requirements that

sometimes conflict with security standards in control system design and operation, e.g., they require authorisation and authentication that should not interfere with or hamper emergency actions. The following table summarises some of these differences and considerations when securing ICS systems (ICS-CERT).

Category	Industrial Control Systems (ICS)	Information Technology Systems (IT)
Availability Requirements	Responses that affect system availability (e.g. rebooting) may not acceptable. Availability requirements may require redundant systems. Outages must be scheduled and planned in advance. Exhaustive pre-deployment testing is required for high availability.	Accept responses such as rebooting. Some system requirements can tolerate availability deficiencies.
Architecture Security Focus	The main focus is to protect edge clients such as field devices and process controllers. It is also important to protect the central server.	The main target is to protect IT assets, including stored and transmitted information. The central server requires extra protection.
Time-Critical Interaction	Emergency interaction and response to human is critical. ICS should have strict access control, but this should not hamper HMI.	Emergency interaction in IT systems is less critical. Access control implementation can be tightly restricted to the required degree of security.
Performance requirements	Real-time. Time-critical response. Modest throughput is acceptable. High delay is not acceptable.	Non-real-time Consistent response. IT systems require high throughput. High delay may be acceptable.
Risk Management Requirements	The primary requirement is human safety, followed by process protection. Fault tolerance is mandatory because even momentary downtime may not be acceptable. The main risk factors are equipment, production, environmental damage or loss of life.	Data confidentiality and integrity is paramount. Fault tolerance requirements are less important in IT because quick downtime is not a major risk. The main risk factor is business operations delay.
Change Management	Software changes must not affect control system integrity. They must be thoroughly tested before being deployed. ICS outages must be scheduled and planned in advance. Control systems may use unsupported Operating Systems.	Software changes are applied promptly using well-designed procedures and security policies. Automated procedures are often used in these systems.
System Operation	Operating systems are possibly owned, systems are often designed without security capabilities. Software vendors must make software changes carefully due to the specific control algorithms, the possible modifications involved in both the software and hardware.	Systems are built for utility with typical operating systems System upgrades are straightforward using automated deployment tools.

Table 2-1 Differences between ICS and IT (ICS-CERT)

2.2.3- Industrial Control Systems Security

Control systems differ from standard IT systems in terms of security goals. Security goals are generally defined as being in one of three categories: Confidentiality, Integrity and Availability (Drias et al., 2015). The security focus of standard IT systems is to protect systems from unauthorized access and to maintain their confidentiality, while ICS developers usually place considerable emphasis on ensuring that systems operate in a safe manner and maintain their functionalities (Availability and Integrity) (Stouffer et al., 2011).

2.2.4- Control System Vulnerabilities

A vulnerability is defined as a root cause of risk that makes an asset unable to resist actions and threats (Grobauer et al., 2011). A vulnerability can be described as a security-related weakness or a flaw in a system design, implementation or configuration that can be viciously exploited so as to harm system security (Grobauer et al., 2011) (Kuang et al., 2006) (Ozment, 2007). These security flaws can be introduced at any phase of control system development lifecycle due to the complexity of application environments and development (Kuang et al., 2006).

Security vulnerabilities could have different structures, times at which they are introduced and extent of associated risk, but they should all be minimised and controlled to reduce any risks that might arise from possible threats. The following gives the conventional meanings of vulnerability, threat and risk (ISO/IEC.27002, 2005).

- **Vulnerability**: is a weakness of any asset in the system that can be exploited by threats.
- **Threat**: is a potential for a vulnerability to become an attack, causing serious harm to the system.
- **Risk**: is a combination of the probability of the incidence of an attack (vulnerability x threat) and its resulting impact.

There are common ICS vulnerabilities that are published and classified by different categories. In 2007, the North American Electric Reliability Council ‘NERC’ (NERC) published the top ten vulnerabilities of control systems as follows (NERC):

1. Insufficient knowledge, procedures and policies that govern the security of control systems.
2. Control system networks are not adequately designed, lacking defence-in-depth mechanisms.
3. Control systems are remotely accessible without appropriate access control.
4. Inadequately maintained system administration mechanisms and software used in ICS.
5. Use of vulnerable wireless communications for control.
6. Use of control system network bandwidth for non-control purposes, using non-dedicated control communications channels for control commands.
7. Inadequate application of tools to discover and report inappropriate activities.
8. Unauthorised devices or applications on control system networks.
9. Unauthenticated control data and commands.
10. Inadequately designed or implemented critical infrastructure.

In 2010, the National SCADA Test Bed (NSTB) identified common vulnerabilities that allow attackers to penetrate ICSs and gain full control of system elements (NSTB, 2016). NSTB published the top ten most critical ICS vulnerabilities based on the likelihood and impact of compromise as follows:

1- Unpatched published vulnerabilities.
2- Using vulnerable remote protocols.
3- Web HMI vulnerabilities.
4- Buffer overflow vulnerabilities in ICS services.
5- Improper authentication.
6- Improper access control (authorization).
7- Using cleartext authentication with standard IT protocols.
8- Unsecured transport of ICS application credentials.
9- Injection and manipulation of control commands and data.
10- SQL injection.

2.2.5- Control System Attacks

Historically, control systems were isolated and operated without any physical connection to public networks (Alcaraz et al., 2012). However, these systems have, over time, become integrated with external networks through their use of services

and data provided by the internet for business purposes (ICS-CERT). This connectivity improves the quality of the services rendered to both customers and operators such as through real-time monitoring, concurrency, peer-to-peer communications, maintenance and redundancy (Alcaraz et al., 2012). As a result, control systems are now susceptible to various kinds of threats (Larkin et al., 2014). Fovino *et al*. categorised attacks into two main classes (Fovino et al., 2010). The first class involves traditional IT attacks that target IT system vulnerabilities. The second class includes ICS-specific attacks that target ICS elements. Fleury *et al*. discussed the following targets for ICS attacks (Fleury et al., 2008):

- System: ICS elements that process critical decisions and calculations.
- User: non-permitted use of user accounts.
- Network: exploitation of communications through IP protocols.
- Process: impacts on control system functions.
- Data: data modification or stealing through unauthorised access.

Fernandez *et al*. categorised attacks according to ICS components as follows (Fernandez et al., 2011):

- Attacks through/against field devices such as malicious alteration of runtime parameters, physical attacks, wrong commands to the field devices and denial of service.
- Attacks through/against field controllers such as malicious of the runtime parameters, wrong commands to the field devices, physical attacks and denial of service.
- Attacks through/against the communication networks such as spoofing, sniffing and denial of service.

From a control engineer point of view, Zhu *et al*. grouped ICS attacks into the following categories (Zhu et al., 2011):

- Invalid input data to controller devices by exploiting network links.
- Inaccurate and misleading output data from controller devices due to exploiting control networks.
- Denial of service – delay or missing task actions.
- Controller historian.

Control systems can be attacked locally via physical access or remotely through unsecured networks. Local access attacks can be gained via different means of entry, as described by Byres *et al*. (Byres et al., 2006) and Anwar *et al*. (Anwar and Malik, 2014):

- Data files such as PLC project files.
- Historian and Enterprise Resource Planning (ERP) servers shared with business users.
- Serial connections.
- Devices such as USB drives and laptops.
- Wireless devices.
- Remote access modems.

Control system attacks could have serious effects on nations and environments. The Stuxnet worm, for instance, caused critical problems at the Natanz fuel enrichment plant in Iran (Langner, 2011). In Australia, ICS attacks caused the Maroochy Shire sewage to be spilled (Abrams and Weiss, 2008). The David-Besse nuclear power plant in Ohio was disabled by the Slammer worm attack (Poulsen, 2003). Duqu (Chien et al., 2012) and Night Dragon (Cyberattacks, 2011) collect information about ICS in order to implement future attacks.

2.2.6- Security Training and Education in Industrial Control Systems

Learning can typically be defined as a continuum process, starting with awareness, building to training and developing, and finally to education (Wilson and Hash, 2003). Figure 2.7 illustrates learning levels in the context of security. The National Institute of Standards and Technology (NIST) defines security awareness, training and education as follows *(Toth and Klein, 2013)*:

Security awareness: is intended to establish recognition of security issues that allow individuals to recognise security concerns in order to reinforce good security practices. Awareness alerts users to vulnerabilities and threats.

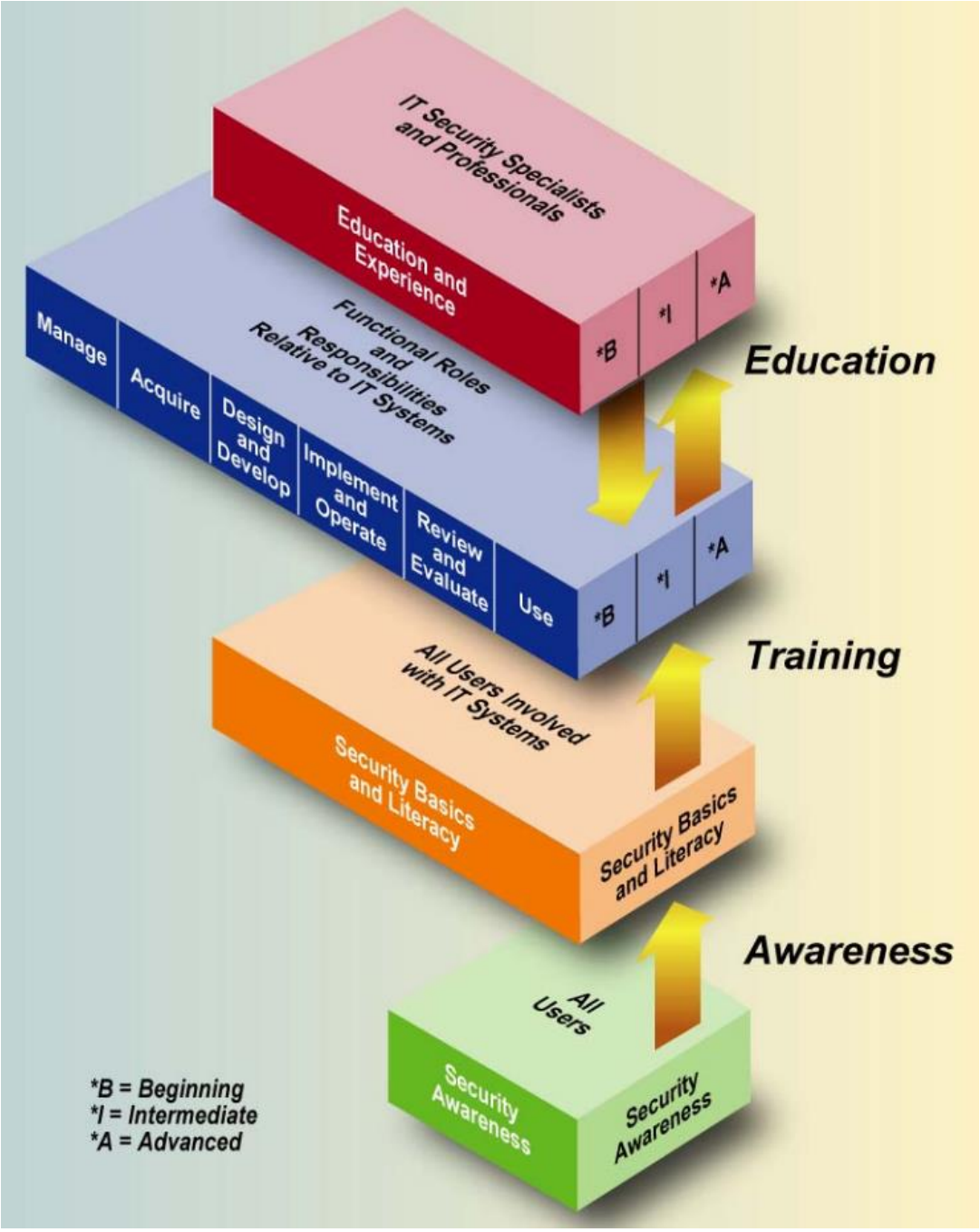


Figure 2-7 The IT Security Learning Continuum (source: reference (Wilson and Hash, 2003))

Security training: is the more formal learning method, which aims to build needed security skills and knowledge. The major difference between awareness and training is that awareness focuses users' attention on an issue, while training aims to teach users skills to perform specific functions.

Security education: is defined by NIST 800-16 as follows:

> *"The 'Education' level integrates all of the security skills and competencies of the various functional specialties into a common body of knowledge, adds a multidisciplinary study of concepts, issues, and principles (technological and social), and strives to produce IT security specialists and professionals capable of vision and pro-active response" (Toth and Klein, 2013)*

2.2.7- Security patterns

In 1994, E. Gamma, R. Helm, R. Johnson, and J. Vlissides, known as th Gang of Four (GOF) among the pattern community, paved the way for design patterns (Gamma, 1995). The main intention of design patterns is to facilitate software development processes by reusing good practice in the design and implementation phase (da Silva Júnior et al., 2013). In the context of security, the first security pattern contribution was published by Yoder and Barcalow, who structured patterns using a GOF template (Yoder and Barcalow, 1998). Security patterns are defined by Markus Schumacher *et al*. as follows:

> *"A security pattern describes a particular recurring security problem that arises in specific contexts, and presents a well-proven generic solution for it. The solution consists of a set of interacting roles that can be arranged into multiple concrete design structures, as well as a process to create one particular such structure."* (Schumacher et al., 2013)

Patterns are typically designed to represent a well-defined solution for a particular frequently encountered problem (Schumacher et al., 2013). They help developers to solve difficult problems by using approved solutions; however, they have only a small impact on system architecture due to certain limitations. Since 2000, the Pattern-Oriented Software Architecture 'POSA' team overcame these limitations and patterns have been used in a number of areas, such as resource management, human-computer interaction, network systems and security (Schmidt and McCormick, 2013).

2.3. Review aims and research questions

The intention behind performing a systematic literature review on industrial control system security is to answer the following research questions:

1- ***What is the state-of-the-art in ICS security by design?***
2- ***What are the challenges of developing secure ICS?***
3- ***What is the current level of security knowledge of control engineers?***
4- ***What is the state-of-the-art in security patterns in ICS?***
5- ***What support and training are related to the design of secure ICS have been proposed?***

2.4. Search and selection process

The literature search was performed through the university library system by searching resources that include databases, journals, conference proceedings and eBooks, as shown in Table 2.2. An advanced search was performed across both the disciplines of both information technology and engineering. Also, the SCADAhacker website (SCADAHACKER, 2016) was searched for related industry articles and contributions such as technical reports and white papers. Each resource was reviewed and the papers that addressed ICS security of any type were identified as being potentially relevant. Each publication was classified as either relevant or otherwise by applying certain inclusion and exclusion criteria, as discussed in the next section.

Search keywords were extracted from each research question. Synonyms, abbreviations and alternative terms were listed with due consideration for subject headlines used in data sources. Keywords were used singularly and in combination to collect data, including: “developing secure industrial control system”, “developing secure SCADA”, “industrial control system security by design”, “industrial control system security challenges”, “SCADA security challenges”, “control engineers’ security knowledge”, “industrial control system security design patterns”, “industrial control system security guidelines” and “industrial control system security training”.

Data Source	Documentation/publisher
Journals	• IEEE Xplore • ACM Digital Library • Springer
Conference Proceedings	• IEEE Xplore • ACM Digital Library • Springer/Lecture Notes in Computer Science (LNCS) • Google Scholar
e-Books	• IEEE-Wiley eBooks Library • Safari Books Online • Springer eBooks • Library search
Grey literature (Technical reports, white papers)	• CPNI • NIST • Deloitte • ENISA • ICS-CERT
Other sources	• Internet

Table 2-2 Selected sources

2.5. Inclusion and Exclusion criteria

Inclusion and exclusion criteria were defined based on the research questions given in Section 2.3.

2.5.1- Inclusion criteria

In order to identify the body of relevant research, sources were measured over two stages (selection-stage-1 and selection-stage-2) against a number of criteria that were defined based on the research questions identified in section 2.3, with respect to the use of different terms associated with industrial control systems (such as ICS, and SCADA, and Automated control systems). Each article must be written in English and meet at least one of the following criteria to be selected:

1- The work relates to industrial control system security development cycles.
2- The work relates to ICS/SCADA security by design.
3- The research investigates the challenges of securing ICS/SCADA.
4- The research relates to study of the role of system engineers in ICS/SCADA security development.

5- The work investigates the level of security awareness and skills of ICS/SCADA developers.
6- The research describes a systematic method for ICS/SCADA security design, such as secure architecture, guidelines, security patterns, and standards.
7- The research proposes training support for the ICS/SCADA workforce.

2.5.2- Exclusion criteria

Any article on the following topics was classified as irrelevant and excluded:

- Papers reporting incident responses.
- Research relating to the study of risk assessment.
- Organisation specific articles.
- Vulnerability analysis.
- Papers proposing safety guidelines.
- Work relating to safety by design.
- Work investigating the relationship between ICS security and safety.

Initially, at ***selection-stage-1***, the selection criteria were applied based on the title and abstract. At this stage, full copies were obtained unless publications were clearly excluded.

Then, final inclusions and exclusions were made after the content was reviewed at ***selection-stage-2***.

2.6. Results

This section shows the results of the search process covering the period from 2008 to November 2016. A search process revealed 379 articles that cover various topics of ICS security, including ICS security requirement engineering, IoT security challenges, ICS security by design, formal verification, vulnerability assessment, threat analysis, incident response, ICS security strategy and governance, and security training. At ***selection-stage-1***, after screening titles and abstracts, 97 articles were felt to be related to the research questions and appropriate for potential inclusion in a systematic literature review.

These articles were then subjected to further screening by obtaining their full texts at ***selection-stage-2***. 32 articles were excluded based on the exclusion criteria.

Data Source	No. of articles selected in ***Selection-stage-1***	No. of articles selected in ***Selection-stage-2***
ACM	16	9
Science Direct (Elsevier)	8	5
IEEE	40	33
Springer	4	1
Grey Literature	25	15
e-books (chapter)	*4*	2
Total	97	65

Table 2-3 Search results

Year of publication	2008	2009	2010	2011	2012	2013	2014	2015	2016 up to November
Number of articles	3	2	4	7	2	10	8	19	10

Table 2-4 Shows the related publications by year (2008-November 2016)

The remaining 65 unique articles identified as appropriate were read thoroughly, as presented in Table 2.3 and Table 2.4. The results of the included articles were collated and summarised (see Table-A1 in Appendix-A) in order to answer the research questions.

2.7. Discussion

The collected data was summarised and classified based on the research questions.

Q1. What is the state-of-the-art in ICS security by design?

In the past, ICSs were developed to meet availability, performance, flexibility, and functional safety requirements that were considered good design goals; in most cases,

this was done without consideration for security issues as in most cases ICSs were isolated from outside networks (Drias et al., 2015) (Shukla, 2016) (Luiijf, 2015). Up to 2008, most of the research effort for protecting ICSs has emphasised reliability (Hadziosmanovic et al., 2012). However, since ICSs have begun to be used to control and monitor critical infrastructures and have been connected over the internet world by adopting IT technologies, security has become a genuine concern for both ICS vendors and owners (Drias et al., 2015) (Durrani et al., 2013) (Fernandez et al., 2011).

The literature shows that developing secure ICS has been the focus of many researchers (Drias et al., 2015) (Axelrod, 2011) (Kunsman et al., 2015). Researchers around the world have considered various aspects of ICS security; the strongest focus can be found in North America, followed by Europe, whilst the Middle East, South America, and Asia are constantly increasing their focus (Hadziosmanovic et al., 2012). Both industrial and government-led research has expended considerable effort in order to enhance the security of ICS over several sectors, e.g., chemical, water, oil and gas.

Hadziosmanovic *et al*. highlighted two reasons for increasing attention on ICS security amongst the research community: first, the importance of ICSs, as they control and monitor critical infrastructure; secondly, the number of ICS security incidents has significantly increased in recent years (Abouzakhar, 2013) (Hadziosmanovic et al., 2012).

Traditionally, ICS developers have focused on safety (Kargl et al., 2014). Pedroza *et al*. distinguished between safety and security engineering as a system that maintains a high level of safety can handle new security threat (Pedroza et al., 2011). They modelled security properties by extending SYSML. Hadziosmanovic *et al*. and Krotofil *et al*. stated that ICSs are, generally speaking, not sufficiently secured and need more directed research effort (Hadziosmanovic et al., 2012) (Krotofil and Gollmann, 2013). The main problem being addressed is in terms of the gap between ICS vendors and information security professionals, vendors mainly focus on the functionality of ICS and lack security knowledge, while security professionals lack experience in ICS in general (Kunsman et al., 2015) (Yang and Zhao, 2014) (Brändle and Naedele, 2008) (Zineddine, 2016). To mitigate this issue, authors recommended

that the two types of firms must bridge this clear gap and work together to implement feasible solutions.

In this context, other researchers recommended that the problem can be solved by systemically integrating security mechanisms across the entire development lifecycle (Fernandez et al., 2011) (Zineddine, 2016) (Oates, 2005). They proposed using security design patterns as a tool to build secure SCADA systems that need to be protected against attacks. Their methodology can be utilised as a guideline for applying the security patterns through all phases of system development. Ur-Rehman and Zivic also introduced a security by design approach (Ur-Rehman and Zivic, 2015). Novak and Treytl illustrated the importance of applying security at various developmental stages and proposed considering security together with safety at the system design phase (Novak and Treytl, 2008). Cheminod *et al*. stated that security by design is the first line of defense in preventing the exploitation of vulnerabilities (Cheminod et al., 2013). Many security issues are recurring problems and can be solved by making security part of the design (Ur-Rehman and Zivic, 2015). A good security architecture will facilitate security implementation within future protected ICSs (He et al., 2016). Otherwise, security flaws might be introduced at different phases of the development cycle (Motii et al., 2015).

In summary, the literature shows that the research effort has significantly increased in the sphere of ICS security and as the focus of security by design. However, they stated that current efforts in this regard are still not sufficient and more effort is needed, especially in the area of ICS security by design.

Many researchers have addressed the fact that ICS lacks security and requires more attention from researchers. The literature demonstrated that including security throughout the entirety of the development lifecycle is paramount to building secure ICS that is resistant to attacks. However, the review also shows that current research focuses on ‘treating’ more than ‘preventing’, and has not achieved effective results in developing secure systems.

Q2. What are the challenges of developing secure ICS?

Protecting control systems against internal and external security threats is one of the great challenges within the ICS domain (Fan et al., 2015). Drias *et al.* conducted a

comprehensive analysis of ICS architectures, focussing in particular on security issues including vulnerability, threat, and security solutions (Drias et al., 2015). Their findings showed that one of the main ICS security challenges is that of developing secure ICS, which can be overcome by tailored security solutions. Kurscheid identified two main challenges: first, applying security is not trivial, requiring greater effort that makes the system more complex; secondly, the true level of system security is hard to demonstrate (Kurscheid, 2013).

The fourth industrial revolution (Industrie 4.0), which is also known as "Smart Factory", has brought further challenges for control system security (He et al., 2016). Sajid *et al*. discussed the security challenges that have, in the main, been inherited from its integration with the Internet of Things (IoT) (Sajid et al., 2016). They focused on developing systems that are simultaneously 'smart' and 'secure', and proposed a security architecture for industrial IoT as a solution. Sadeghi *et al*. also discussed security challenges related to industrial IoT and proposed solution of secure engineering to counter associated security risks (Sadeghi et al., 2015).

On the other hand, many researchers have highlighted the challenge of understanding ICS security (Kunsman et al., 2015) (Annex, 2011) (Luiijf, 2015). ENISA identified 'creating a security culture' as a key challenge for securing control systems (Annex, 2011). Amaechi and Counsell highlighted the lack of clarity and understanding of security as being a key challenge to system design (Amaechi and Counsell, 2012). Annex also indicated the challenge of insecure ICS by design and the lack of proper governance of ICS; they proposed a number of recommendations to improve such governance by raising awareness of security (Annex, 2011).

In summary, the literature addressed the various challenges associated with securing control systems based on different aspects. However, the challenges in our sphere of interest fall into two categories:

- ICS security by design
- ICS security education and awareness

The literature also shows that other challenges of securing ICS such as lacking tailored methods that take the nature of ICS into account, following by the the affect of implementing security on system performance and the lack of impact assessment methods.

Q3. What is the current level of security knowledge of control engineers?

ICS Security awareness and education has become a real concern (Durrani et al., 2013) (Amaechi and Counsell, 2012, Boyes, 2015) (Miyachi and Yamada, 2014) (Vaughn Jr and Morris, 2016). Amaechi *et al*. and Savola *et al*. indicated the lack of security awareness and skills in the ICS domain and recommended using security guidelines to develop secure systems (Amaechi and Counsell, 2012) (Savola and Ahonen, 2006). Pauna indicated that the current security challenges imply the need for ICS professionals with good security knowledge (Pauna et al., 2014). Security-unaware developers and employees were identified by Durrani *et al*. as being the weakest link in terms of system security (Durrani et al., 2013). The lack of security awareness and training was outlined as being amongst a set of ten security concerns associated with ICS (Vaughn Jr and Morris, 2016). The review by Graham *et al*. identified six main factors that are root causes of ICS security vulnerabilities, one of which is the lack of security training (Graham et al., 2016). Axelrod outlined the knowledge gap between ICS professionals and information security professionals, and indicated the need to increase security training and education (Axelrod, 2011). Ismail *et al*. conducted interviews with ICS professionals from different countries to measure their levels of security awareness (Ismail et al., 2014); their findings showed that organisations, generally speaking, lack security awareness and training.

ENISA highlighted the importance of security education and awareness in creating a security culture that can overcome the challenge of developing secure ICSs (Annex, 2011). A security awareness program is paramount for mitigation and appropriate defence plans (Durrani et al., 2013). Amaechi and Counsell investigated design success factors using ICS as a case study (Amaechi and Counsell, 2012). They found that raising security awareness using design methods and learning materials can overcome the risk associated with lack of knowledge. Similarly, Miyachi and Yamada stated that the ICS community, including developers, operators, owners and users, should have a level security knowledge as appropriate to their responsibilities (Miyachi and Yamada, 2014). Security training and awareness are the main aspects

needed to create a security culture that can mitigate ICS vulnerabilities (Navarro et al., 2014) (Annex, 2011).

Pauna indicated the challenge of developing the proper security education relating to operational issues, as ICS and information security are very different topics (Pauna et al., 2014). There are many educational programs in research laboratories. Mississippi State University has a strong focus on ICS security, and there is a security course introduced by Luallen and Labruyere for control system developers (Luallen and Labruyere, 2013). Other organisations have put considerable effort into security training and education such as CPNI, NSTB, SANS, NERC and ICS-CERT (ICS-CERT) (NERC) (NSTB, 2016) (CPNI, 2016) (SANS, 2016) (Francia III, 2011).

In summary, security education is typically offered by colleges and universities that provide a degree when obtaining the associated learning program. The literature clearly indicates that security awareness and training plays a major role in improving ICS security (Wilson and Hash, 2003) (Stouffer et al., 2011). Control engineers should receive security training that focusses on their responsibilities in order to understand organisational policies, security weaknesses, recommended security patterns and how to properly protect ICS resources (Stouffer et al., 2011) (Permann et al., 2006). Unfortunately, the previous literature also indicates that ICS developers lack security awareness and knowledge, and there is a culture gap between developers and the security experts responsible for providing security solutions to protect these systems from attack (Stouffer et al., 2011).

Q4. What is the state-of-the-art in using security patterns in ICS?

Security patterns are a feasible tool for reducing design flaws in a system (Ur-Rehman and Zivic, 2015). Using security patterns in control systems was first proposed in reference (Fernandez et al., 2011). The authors of this article provided a mechanism to apply security patterns throughout the whole development lifecycle. Motii *et al.* demonstrated a guideline for selecting security patterns using a control system case study (Motii et al., 2015). Ur-Rehman and Zivic proposed a security by design approach using security patterns (Ur-Rehman and Zivic, 2015). SANS also proposed ICS security architecture using security patterns.

In summary, using security design patterns helps in the reuse of expert knowledge and the mitigation of vulnerabilities introduced during system design (Ur-Rehman and Zivic, 2015). However, the literature clearly illustrates that very little research effort has been expended on using pattern-based design approaches within the ICS domain (Fernandez et al., 2011) (Motii et al., 2015) (Obregon, 2015).

Q5. What support and training related to the design of secure ICS has been proposed?

ICS developers lack methods and tools that support ICS security engineering (ICS-CERT) (Brändle and Naedele, 2008) (CPNI, 2016). Brundle and Naedele recommended supporting ICS developers by providing security training with a level of abstraction without complex security details (Brändle and Naedele, 2008). Security guidelines can help ICS developers to apply security throughout the development lifecycle (Motii et al., 2015). Homeland Security recommended that vendors should educate developers in secure coding and best practice in order to detect vulnerabilities during the system development lifecycle, and, of course, before system release (Nelso and Chaffin, 2011). While they focused on secure coding, supporting programmers and securing implementation provide a sound approach for other developmental phases.

Both the research community and government organisations have published numerous articles on common ICS vulnerabilities and proposed security measures and solutions and training programs for control systems (McGrew and Vaughn, 2009) (Stouffer et al., 2011) (ICS-CERT). For example, the Department of Homeland Security delivers critical infrastructure cyber-security training from Idaho National Laboratories (INL) in the US. The Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) also offers a number of courses with up to seven days security training (ICS-CERT training). The National SCADA Test Bed (NSTB) (NSTB, 2016) delivers three levels of SCADA security courses namely the introductory, intermediate and advanced (NSTB, 2016). While these training programs are valuable, they are offered for a limited amount of time and number of attendees.

In addition, there are a number of tools that have been developed to support ICS security. Homeland Security has built CSET, which evaluates system security against

certain standards using a question and answer method (ICS-CERT). Pedroza *et al*. proposed a tool that extends a popular modeling language (SYSML) to support system designers in modeling security properties (Pedroza et al., 2011). However, they did not focus on security awareness and education.

The Cyber Security Modelling Language (CySeMoL), which has been created by the KTH in Stockholm, estimates the probability of success of attack. CySeMoL does not provide information about existing weakness or possible solutions, however, and additionally (though perhaps understandably) only the tool creators can update it.

ValueSec, which has rebuilt Lancelot, is a risk management platform (Prez and Machnicki, 2013). It is used to analyse security risks within the SCADA environment. While this tool performs security analysis and suggests mitigation plans, it does not focus on the security learning aspect.

Francia *at al*. reviewed ICS security best practices and risk assessment; their research findings indicated that most of the currently available tools do not focus on the learning aspect. Tools mainly provide security awareness without training and education (Foo et al., 2013).

In summary, the literature clearly shows that ICS developers lack security support in terms of both technical (within the system development environment) and knowledge support through training programs. The literature also indicated the importance of the learning aspect of any proposed method.

This systematic review revealed a number of key references that identified the academic and industrial motivation behind this research:

- Developing secure control systems has become important as they control and monitor critical infrastructure, in addition to the significant increase in the number of ICS security incidents over the last decade (Hadziosmanovic et al., 2012). Therefore, there is a need for greater research effort focussing on ICS security.

- Understanding ICS security was identified as a key challenge in system design (Annex, 2011) (Amaechi and Counsell, 2012). Security awareness and training

plays an important role in improving ICS security (Stouffer *et al.*, 2011). However, there is a culture gap between ICS vendors and information security professionals, as vendors mainly focus on the functionality of ICS and lack security knowledge, while security professionals lack experience in ICS (Axelrod, 2011) (Yang and Zhao, 2014).There is a need to increase security training and education to bridge this gap and support developers in working together with security professionals to develop feasible solutions.

- The Security for Industrial Control System (SICS) Framework, provided by CPNI, presents good practice principles for ICS security (ICS-CERT) (CPNI, 2016). The framework identifies security awareness and skills as one core elements in ensuring ICSs are secure by design. Therefore, it is imperative to develop a powerful training method to inform improvements of ICS developers' security knowledge.

- Well-structured solutions such as security patterns can be used as a basis for developing security guidance and good design practice within the system development cycle (Fernandez et al., 2011). These patterns capture security expertise by identifying both a security problem and its solution. The pattern-based approach is believed to have the potential to solve the problem addressed in this research and can be adopted to effectively enhance security knowledge of system developers.

- Security patterns provide a possible description as to how to solve problems in the form of worked solutions. However, security patterns have been criticised as they do not provide a practical guide as to how to they can be selected or applied (Ur-Rehman and Zivic, 2015). Therefore, there is the need for a method that can guide system developers in their selection and reuse of these patterns.

2.8. Conclusion

This chapter presented a systematic literature review of ICS security by finding answers from the research literature to five research questions. Most of the articles studied identified the limitations of the current research, the lack of security awareness and skills among ICS developers, and recommended applying security at

all development phases. However, little attention was given to pattern-based approaches and developing methods that can support ICS developers within the system development cycle.

Based on the above discussion, there is a clear need for a systematic method of considering security requirements early in ICS development phases. It was also found that there is a need to pay more attention to the role of system developers in building secure ICSs, and in improving their security knowledge. This gap identifies the roadmap for our research. In particular, pattern-based approaches, ICS security by design, and tailored training approaches, will be adopted to support developers in building secure control systems.

The next chapter discusses the selected research methodology, emphasising the research paradigm, mapping this research into the selected methodology, and data collection and data analysis methods that will be used in this research.

Chapter 3
Research Methodology

Chapter objectives

- To discuss information system research paradigm candidates
- To justify the selection of the Design Science Research (DSR) methodology
- To discuss the phases of the Design Science Research methodology.
- To introduce our research design in line with the selected methodology

3.1. Introduction

This chapter presents the research methodology used to carry out this research. The chapter explains and justifies the selection of a fitting research approach, employing, and adhering to the guideline of the chosen research methodology.

It was necessary to ensure that our research followed a clearly defined path through research approaches and methodologies, as explained by Kumar:

> *"Research methods means all those methods and techniques that are used for conducting research, and thus refers to the methods the researcher uses in performing research operations"*(Kumar and Phrommathed, 2005).
> *"Research methodology is a way to systematically solve the researcher's problems; it may be understood as the science of studying how research is done scientifically"* (Kumar and Phrommathed, 2005).
> *"The research approach is that the researcher should himself pose a question and procedures for throwing light on the questions concerned for formulating or defining the research problem"* (Kumar and Phrommathed, 2005).

Therefore, according to these definitions, it was important to distinguish between research paradigms and methodologies in order to select an appropriate method to design this research.

This chapter is divided into the following sections. Section 3.2 discusses information system research paradigms and explains the appropriate paradigm for our research. Design Science Research (DSR) methodology was selected, and is discussed in

Section 3.3. Section 3.4 introduces our research design and demonstrates how it was mapped into the work packages of the selected methodology.

3.2. Selecting a Fitting Research Methodology

The nature of information system research is complex as it gains its contributions from multidisciplinary research fields such as mathematics, engineering, behavioral science and natural science (Galliers, 1992). There are a variety of research approaches, paradigms, methods, and techniques that can be used in different research contexts (Al-Debei and Fitzgerald, 2009). Thus, selecting an appropriate research method is a key task during the research design process.

A paradigm is defined as a set of philosophical perspectives, assumptions, and guidelines that guide the activities researchers carry out during the research process (Mingers, 2001) (Denzin and Lincoln, 2011).

The paradigms of information system research have certain key characteristics that can be classified into three fundamental categories (Orlikowski and Baroudi, 1991): Firstly, it has an ontological character, based on the empirical world whether it is 'objectively' independent of human observers, or 'subjectively' considers human actions and beliefs. The second character is epistemological, raising many questions including what can be known? how can knowledge be created and evaluated? or what is the relationship between the knower and the knowledge? The final character is that of methodology, which represents the relationship between theory and practice. It identifies the strategic approach as to how researchers can carry out their research and gain knowledge, rather than use particular techniques and data analysis.

Across these characteristics, there are different views of what research actually is, and how it relates to the developed knowledge. Research paradigms guide researchers in making decisions and carrying out research. The awareness of the whole range of research approaches, paradigms and strategies are beneficial as such understanding normally supports an informed selection (Orlikowski and Baroudi, 1991). Therefore, the next subsections will discuss the research paradigms in information systems to guide the selection of the research methodology towards that most appropriate for carrying out this research. The main four paradigms in information research are classified as positivist, interpretive, critical and design

science (Orlikowski and Baroudi, 1991) (Chua, 1986) (Klein and Myers, 1999) (Von Alan et al., 2004).

3.2.1. Positivist Paradigm

Positivism was defined by Cooclian as a "scientific method". This paradigm was claimed by the French philosopher, Auguste Comte (1798-1857), who employed it in social science research, and demonstrated that observation and reason could be used to understand human behaviour (Coolican, 2014). The research can be categorised as positivist if it gives evidence of containing a hypotheses, measures research variables either operationally or quantifiably, tests the formulated propositions and provides conclusions about a phenomenon based on a sample of the research population (Orlikowski and Baroudi, 1991). The reality is objectively given and discovered using measurable factors that are independent of researchers and participants (Oates, 2005) (Myers, 1997). The positivist paradigm adopts the methods of natural science as an approach to producing knowledge about human society (Cohen et al., 2013). One of the key criticisms of positivist research that it ignores its social environment, and as a result neglects important meanings (Collis and Hussey, 2013).

A positivist paradigm may not be appropriate to this research because it aims to predict and clarify external reality (Orlikowski and Baroudi, 1991), while our research aims to construct a reality. In addition, a positivist paradigm employs observation, and quantitative, or statistical, methods to achieve research aims (Orlikowski and Baroudi, 1991). By contrast, these methods do not support the main aim of this research, which is that of developing an effective supportive method that can assist developers in designing control systems.

3.2.2. Interpretive Paradigm

Research can be classified as interpretive if it aims to understand the information system context and the process whereby it effects, and is effected by, the context (Cohen et al., 2013). Interpretive research assumes that the knowledge of reality is formed by its social context and obtained through social constructions such as language, tools and documents (Cohen et al., 2013). Reality is interpreted by

individuals and their subjective meanings through interaction with the social environment. In other words, interpretive research aims to understand beliefs and interpretations in a context appropriate to the generation of meanings, and describing and explaining phenomena through participation and qualitative methodology.

The interpretive paradigm has been subject to criticism by a number of researchers (Cohen et al., 2013) (Bernstein, 2011, Fay, 1987, Gibbons, 1987). They addressed different drawbacks associated with interpretive research, such as missing external circumstances and being ignorant of historic changes.

Our research may not be an interpretive research due to the following distinguished characteristics: (1) ***Research aim:*** this research aims to change the state of the security knowledge situation by improving security awareness and skills within the control engineers' community, unlike the interpretive research, which aims to understand, describe, explain, and interpret a phenomenon. (2) ***Epistemological character:*** in this research, knowledge is developed through the construction of a new supported training method. By contrast, the knowledge developed by interpretive research emerges from participants' interactions.

Despite the above differences between this research and interpretive research, this research does employ one of methods of interpretive research, namely that of the qualitative method, to support research problem identification, as explained later in section 3.4.1.2.

3.2.3. Critical Paradigm

Critical research constructs reality based on a historical perspective by social, economic, political, and cultural forces that have been created or shaped over time by individuals (Myers, 1997). Critical research aims to enhance the opportunity to realise human potential by helping to reduce the causes of unwarranted domination through a social critique (Avison et al., 2008). The critical paradigm and interpretive paradigm share a number of research characteristics, they support each other, and employ methods that are compatible with both kinds of research (Khazanchi and Munkvold, 2003).

This research may not be critical because security knowledge is not created by the facts of historical practice. Our research develops knowledge through building an adaptable security supported tool. The second key criticism is that critical research employs investigative methods to measure beliefs and assumptions. These methods cannot support the main aim of this constructive research, which is that of developing a supported method and measuring its artefactual impact on control system developers.

3.2.4. Design Science Paradigm

The design science paradigm was first defined by Walls et al. (Walls et al., 1992) Later, in 2004, Hevner et al. refined the definition of this paradigm and presented an approach based on seven guidelines (Von Alan et al., 2004). The paradigm started to emerge and be used in information system research to enhance the relevance of the information system discipline (Purao, 2002, Vaishnavi and Kuechler, 2004). Design science research aims to develop novel artefacts in order to improve social or organisational systems (Von Alan et al., 2004).

The design science paradigm is appropriate to our research because of the similarities in its research characteristics, as follows: (1) ***Research aims:*** design science research aims to construct a reality by changing the state of the world situation. Similarly, this research aims to change the situation of control system security by improving developers' security awareness and skills through supported methods. (2) ***Epistemology:*** in design science research, knowledge is created through making. Likewise, this research contributes to knowledge by developing an adaptive security supported method that contributes by assisting control system security by design. (3) ***Methodology:*** the methodological models of design science research paradigms are compatible with the aim of this research as they are developmental and capable of measuring the constructed artefact. In addition, this paradigm may advance the aims of our research by helping the researcher to scientifically understand one of the research problems and provide an innovative solution with the further opportunity to examine its feasibility and effectiveness using associated evaluation methods. Therefore, the design science approach is deemed appropriate and more consistent with the purpose of our research.

3.3. Design Science Research (DSR) Methodology

Design Science Research (DSR) is defined as a problem-solving paradigm for information system research, which aims to create innovative artefacts that define products, practices, ideas and technical capabilities through analysis, design, implementation, and management (Von Alan et al., 2004). DSR can be described as formulating design theories to solve a particular problem by developing artefacts including constructs, methods, models, human-computer interfaces, algorithms, or other artefacts (Walls et al., 1992) (Venable, 2006) (Gregor and Jones, 2007). The literature has also shown some conditions and research missions that a research project has to fulfil to be classified as DSR. von Alan *et al*. (Von Alan et al., 2004) presented practical guidelines and argued that DSR should:

1. Produce an applicable artefact such as construct, method or model.
2. Develop technology-based solutions for relevant real-world problems.
3. Demonstrate efficacy, quality, and utility of the design through well-executed evaluation methods.
4. Produce a contribution through both the form of the artefact and the knowledge base of the design.
5. Apply a methodology to construct and evaluate the artefact.
6. Present the research results to technology- and management-oriented audiences.

3.3.1. Design Science Research Process

Vaishnavi *et al*. (Vaishnavi and Kuechler, 2015) introduced the process of DSR, starting with identifying a real-world problem and ending with appropriate conclusions, as shown in figure 1. ***The first phase*** defines a relevant problem that may derive from reviewing related work. ***The second phase*** uses this knowledge base to find and suggest feasible solutions to the problem being addressed. It is essentially a creative step wherein a formal proposal is produced based on a novel configuration. ***The third phase*** is developing the suggested solutions to construct the artefact. ***The fourth phase***, once the artefact is constructed, is to evaluate it according to criteria that are usually made explicit within the research proposal. Evaluation is performed through testing the utility of the artefact, its usefulness, and its applicability. ***The fifth phase***, is the final phase of a research effort that typically includes a thesis write-up.

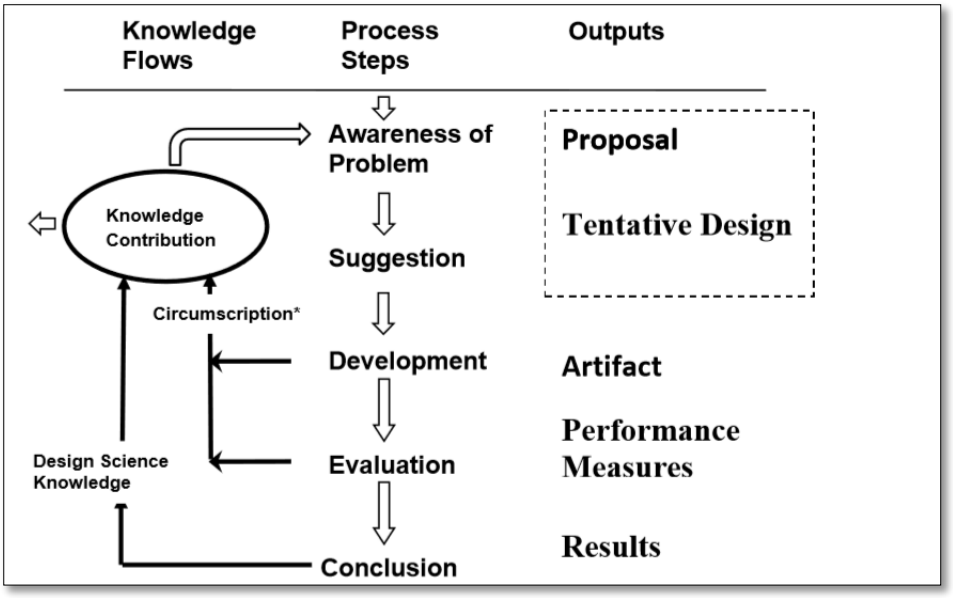


Figure 3-1 Design Science Research Process model (source (Vaishnavi and Kuechler, 2015))

The design science research methodology was selected to conduct our research as it is highly relevant to information technology research. This methodology supports a paradigm of pragmatic research that creates innovative artefacts to solve real-world problems (Von Alan et al., 2004, Simon, 1996). This research is fundamentally constructive as it constructs a new, supported method. It aims to extend human capabilities, namely developers' security knowledge, and achieve the desired outcomes by creating an innovative artefact. Thus, DSR methodology reaches to the core of what has been constructed, applied, evaluated, and improved upon in our research.

The following section demonstrates how this research is designed in line with the design science research methodology and discusses the associated methods that were used to achieve the goals of this research.

3.4. Mapping this research into a design science research model

According to the DSR process, our research methodology consists of five work phases that interact with each other within the research process, as shown in figure 3.2.

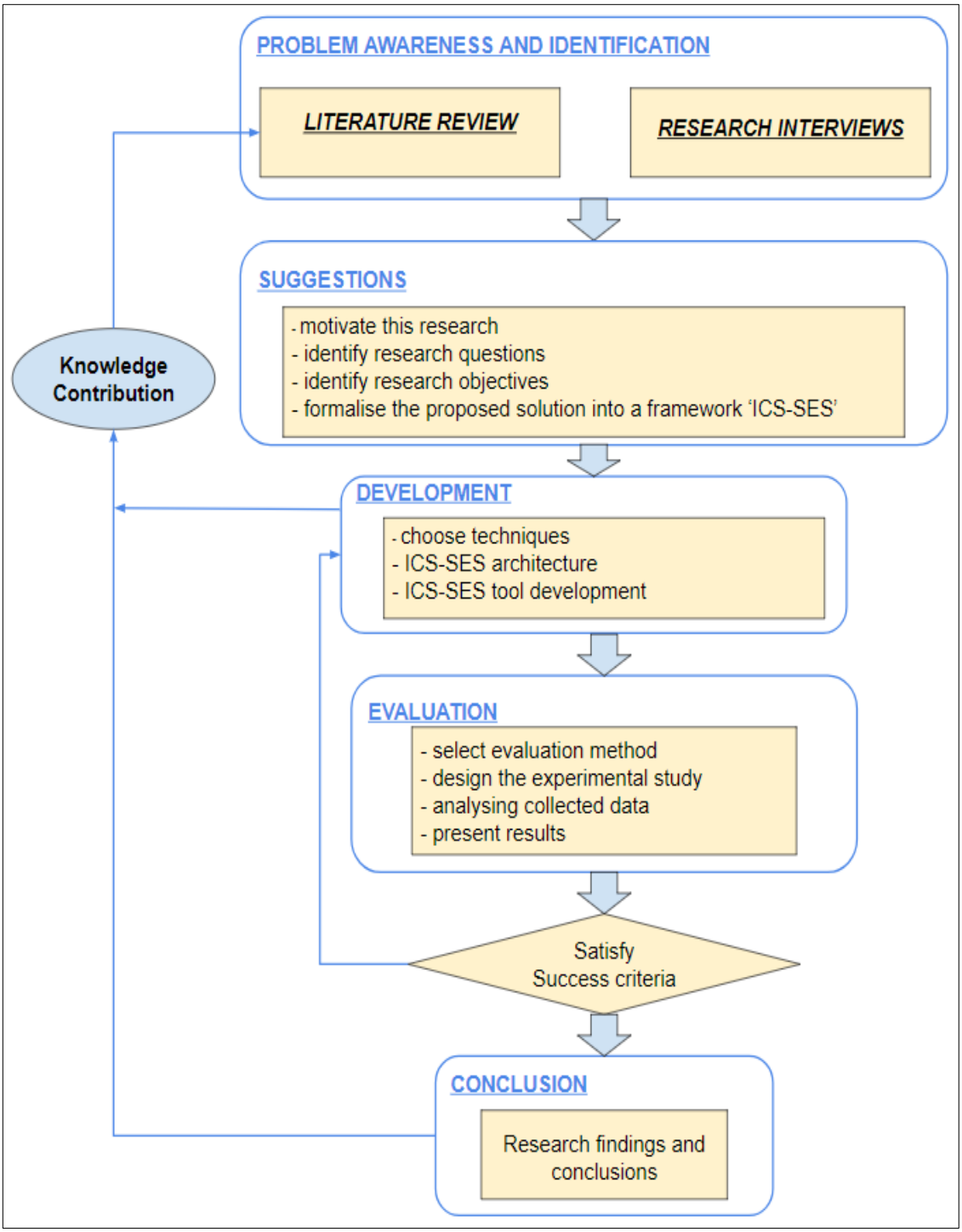


Figure 3-2 This research methodology, in relation to DSR methodology

3.4.1. Problem Awareness phase

This research project started by addressing a problem in industrial control system development environments. A search on the related literature was carried out to identify the knowledge gap and understand the research problem. This stage is also

known as problem investigation, where information about the problem is collected and understood without yet solving it (Simon, 1996). The problem investigation process was classified in reference (Wieringa, 2009), based on its emphasis, into four categories, each of which leads to different views of the problem identification process:

Problem-driven investigation: where problems need to be diagnosed before solving them. This investigation describes a phenomenon, formulates and tests hypotheses as to the causes of the problem and identifies its priorities.

Goal-driven investigation*:* there is no problem experienced or that needs to be identified, but nevertheless there are reasons to change the world in order to reach some goal. This kind of investigation describes and operationalises stakeholder goals, and defines goal priorities.

Solution-driven investigation*:* where a known solution is applied to new problems. This investigation includes identifying a new functionality and the utility of existing technology for solving new problems.

Impact-driven investigation*:* this is also called the evaluation phase, where the outcome of past solutions is evaluated. This investigation identifies and explains the impacts of previously implemented solutions.

In this research, the problem identification falls under the category of being a problem-driven investigation, which is conducted through reviewing related literature and conducting research interviews with domain experts, as shown in Figure 3-3. These studies give the researcher insight into the problem that control systems lack security engineering.

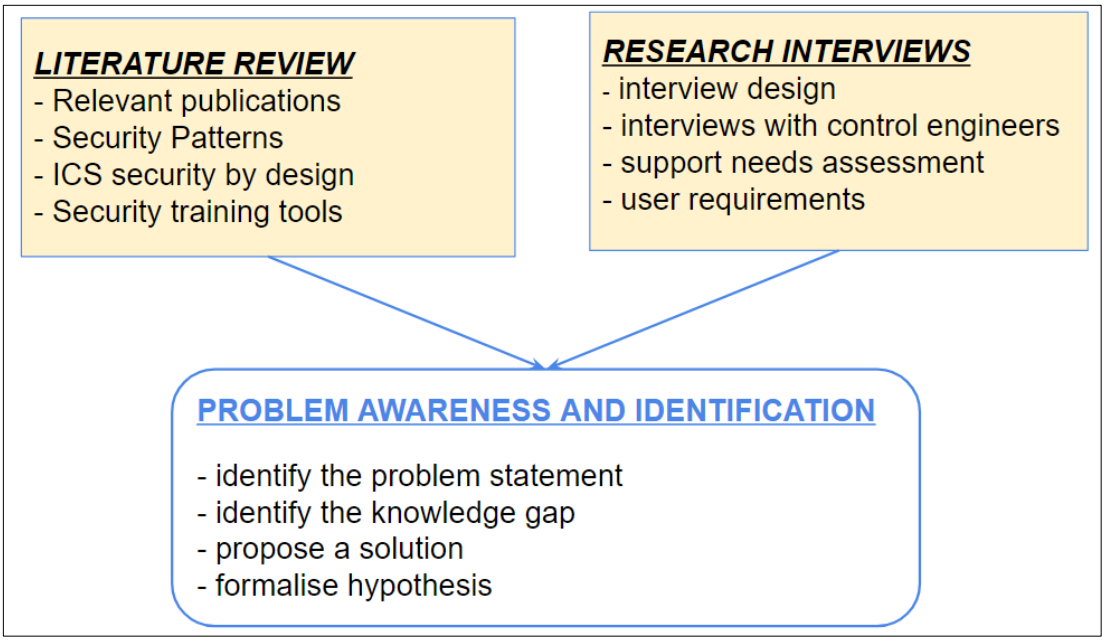


Figure 3-3 Research problem awareness and identification

3.4.1.1. Systematic Literature Review.

A systematic literature review was conducted to help the researcher recognise the relevant aspects of the research, which in any case is highly recommended in DSR guidelines (Von Alan et al., 2004). A systematic review is "*a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest*" (Kitchenham et al., 2009).

The researcher decided to conduct a systematic literature review due to the associated advantages highlited by reference (Kitchenham et al., 2009) as the following: 1) it less likely that the results of the review are biased. 2) it provides rich information from findings across a wide range of previous studies, which provides consistent results as to evidence of the robust nature of the phenomenon or otherwise inconsistent results that can be further studied. 3) It is conducted according to a predefined search strategy that allows a researcher to synthesis existing work in a thorough and fair manner. However, systematic reviews require considerably more effort than conventional literature reviews.

Our systematic literature review was based on the guidelines defined by reference (Kitchenham et al., 2009). The review consists of three main stages: the planning, conducting and reporting stages.

In the planning phase, the research questions were defined for the review. The review protocol was also developed and evaluated. It was necessary to predefine the review protocol to reduce the possibility of research bias. The components of the protocol included the following elements: background, review aims and research questions, study selection criteria, and data extraction. Since this review was a part of PhD project, the protocol was reviewed by the researcher's supervisors, as recommended in reference (Kitchenham et al., 2009).

In the second phase, the review was conducted by applying inclusion and exclusion criteria, extracting information, and collecting data. The review was conducted to find as many related articles in the literature as possible. A list of sources was obtained including references, journals, conference proceedings and industrial websites. The sources were thoroughly searched based on our selection criteria. Data was collected in relation to the research questions.

The last phase of the review included reporting and evaluating the results. The results of this review were reported and discussed in Chapter 2.

3.4.1.2. Research interviews

The design science research methodology can encompass methods from other paradigms such as using positivist paradigm methods for the evaluation process, or using interpretive paradigm methods for problem identification and other fundamental requirements (Weber, 2010).

In this research, a number of interviews were conducted with control engineers to enrich the findings of the systematic literature review. The use of the interview method was deemed useful as it was expected to improve the understanding of the research problem. The interviews inclemently enhanced the knowledge gained from the literature by investigating the current security knowledge of control engineers, their need to design secure systems, existing security support, and security training methods. To this end, the research interview was designed to support awareness of the problem and assess engineers' needs in order to develop control system security by design, as introduced in Chapter 4.

The three main types of research interviews are semi-structured, structured and un-structured interviews (DiCicco-Bloom and Crabtree, 2006). The use of semi-structured interviews was preferred in this research as opposed to the other types because they are better suited to small samples to derive supplementary information (Laforest, 2009). In addition, semi-structured interviews keep the researcher focussed on the main aspects of the problem under consideration, while at the same time allowing the researcher to be open to any new ideas that may emerge during an interview process (DiCicco-Bloom and Crabtree, 2006).

Seven semi-structured interviews were conducted, recorded, transcribed and qualitatively analysed, as explained in Chapter 4.

The data collected from both the literature review and research interviews, which are presented in Chapters 2 and 4, respectively, gave considerable insight into the research problem and helped to derive a novel solution of forming an adaptive security supported method. As a result, the research hypothesis was formalised and the research questions were set. However, the hypothesis was continuously evaluated and adjusted during the research process, as it represents the results of the entire body of this research.

3.4.2 Suggestions phase

Following the previous phase, the suggestion step is intended to allow a proposed solution in a tentative framework to be formalised (Offermann et al., 2009). This phase is essentially a creative step wherein a new security supported method was envisioned based on the use of a tailored on-the-job training approach.

Our research presented a security supported framework, named ICS-SES, that can be configured into a development environment to be used by control system developers during their work to assist them in designing secure systems and, consequently, enhance their security awareness and skills, which is the main goal of this research. During this phase, the framework was produced by identifying its components and specifying their interrelationship, as discussed in Chapter 5.

3.4.3 Development phase

After identifying the problem of the lack of security knowledge and proposing a security supported solution, this solution then needs to be developed and evaluated in order to evaluate the outcomes of the entire research (Offermann et al., 2009). Development is a creative engineering process (Offermann et al., 2009).

In this research, the intention of the development phase was to develop the proposed supported method in the form of an artefact, including the design and implementation processes. It was concerned with translating the proposed conceptual framework into an implemented tool, this process being called 'forward engineering' (Fernández-López and Gómez-Pérez, 2002). This led to deciding which software development methodology would be undertaken and which techniques would be used.

Software development methodology is defined by the IEEE society as *"the application of a systemic, disciplined, quantifiable approach to the development, operation, and maintenance of software*" (Radatz et al., 1990). The literature presented a number of development methods such as the waterfall, spiral, formal, agile and prototype methods, which are employed based on the type of project, changes in requirements, project size, complicity, etc. (Farrell, 2007) For example, the waterfall method is typically used when user requirements are stable and unchanging during system development (Bassil, 2012). Amongst the available development methodologies, the prototype method was chosen for developing the security supported tool because it provides an early view of functionality, which supports the process of changing and refining requirements if so needed (Bischofberger and Pomberger, 2012). The prototyping takes an iterative approach, which breaks the project into small segments and builds a prototype for each (Bischofberger and Pomberger, 2012). It is based on developing, examining, analysing, and refining the prototypes until they meet the appropriate requirements, which in our research are defined in the success criteria in Chapter 1, Section 1.5.

A prototyping tool for the ICS-SES method was implemented and used in an evaluative experimental study, as explained in Chapter 6.

3.4.4 Evaluation phase

After clearly defining the research question and the constructed artefact reaching a sufficient state, it was necessary to select a suitable evaluation method (Vaishnavi and Kuechler, 2004). The evaluation phase is an important requirement in design science research (Cleven et al., 2009). The quality and efficacy of an artefact must be demonstrated via well-executed evaluation methods (Hevner and Chatterjee, 2010).

Evaluation can be achieved by means of action research, by conducting case studies, surveys, lab experiments or through simulation (Offermann et al., 2009). In design science research, the contribution is mostly the artefact itself, and hence it must be clearly validated and identified as a new research contribution (Hevner and Chatterjee, 2010). Artefacts can be assessed in terms of functionality, utility, performance, and other relevant quality attributes (Hevner and Chatterjee, 2010). In fact, the evaluation phase is intended to answer the question "*How well does an artefact work?*" (March and Smith, 1995). Evaluation can also provide essential feedback about the quality and design during the development phase (Hevner and Chatterjee, 2010).

According to the evaluation guidelines presented by Hevener *et al.*, effective evaluation requires the appropriate use of research methodologies from the knowledge base (Hevner and Chatterjee, 2010). Table 3.1 shows design science research evaluation methods as summarised by reference (Hevner and Chatterjee, 2010). The use of one or more of the methods reported in table 3.1 can help the researcher to convince the research community as to the value and validity of the proposed solution (Vaishnavi and Kuechler, 2015).

1. Observational	*Case Study – Study Artefact in Depth in Business Environment*
	Field Study – Monitor Use of Artefact in Multiple Projects
2. Analytical	*Static Analysis – Examine Structure of Artefact for Static Qualities (e.g., Complexity)*
	Architecture Analysis – Study Fit of Artefact into Technical IS Architecture
	Dynamic Analysis – Study Artefact in Use for Dynamic Qualities (e.g., Performance)
3. Experimental	*Controlled Experiment – Study Artefact in Controlled Environment for Qualities (e.g., Usability)*
	Simulation – Execute Artefact with Artificial Data
4. Testing	*Functional (Black Box) Testing – Execute Artefact Interfaces to Discover Failures and Identify Defects*
	Structural (White Box) Testing – Perform Coverage Testing of all Execution Paths in the Artefact

Table 3-1 Design Evaluation Methods

(source: reference (Hevner and Chatterjee, 2010))

Experimental methods play an important role in software engineering evaluation as they allow researchers to contribute to the body of knowledge through observation and empirical evidence (Basili, 2007). They provide a scientific basis for software engineering (Wohlin et al., 2006). Experiments can be controlled experiments, where subjects are randomly assigned to different treatments, or quasi-experiments, which are used when random assignment cannot be performed (Wohlin et al., 2006).

Controlled experiments provide the most rigorous evidence of any correlation relationships between the research tool and the outcomes (Hevner and Chatterjee, 2010). They are highly controlled as they are based on fixed designs and more formal procedures than other empirical methods (Whitten, 1990). This advantage allows researchers to plan and design rigourous experimental studies that ensure a high degree of validity (Wohlin et al., 2006). Controlled experiments are often used when researchers need to evaluate changes in participants' knowledge, skills, and behaviour (Wohlin et al., 2006). They are also used to demonstrate the effectiveness of a treatment through the application of statistical methods (Hevner and Chatterjee,

2010). Therefore, the controlled experiment method was selected to evaluate the usability of the educational tool in terms of effectiveness, efficiency, and ease of use.

The experiment study design was based on the practical guidelines provided by reference (Ko et al., 2015) to allow the evaluation of software engineering tools with human participants in line with our research question ***"Can a supported tool assist developers in designing secure control systems?"***, as introduced in Chapter 6. The results obtained are analysed and discussed in Chapter 7.

3.4.5 Summarising Results and Drawing Conclusions

The last phase of the research process is intended to summarise the research findings and draw appropriate conclusions, which clearly identify research contributions, and publish them in the form of PhD thesis, or conference or journal articles (Offermann et al., 2009). The fundamental assessment for any research is "*what are the new contributions*?" The authors of reference (Hevner and Chatterjee, 2010) stated that:

> *"Effective design science research must provide clear contributions in the areas of the design artefact, design construction knowledge, and/or design evaluation knowledge."* (Hevner and Chatterjee, 2010)

The design artefact contribution is the artefact itself, and will be such as a new tool, model or method that must be clearly evaluated and identified as a research contribution (Hevner and Chatterjee, 2010). The theoretical foundation's contribution is one of extending and improving existing theoretical foundations within the knowledge base of the research by the creative use of a new construct (Hevner and Chatterjee, 2010). The evaluation methodology's contribution is the creative use of a new evaluation method (Hevner and Chatterjee, 2010).

The complete results of this research are published in a form of PhD thesis. The research findings were summarised and discussed in Chapter 8 . The original and complementary research contrubutions were outlined. New directions for future work were also suggessted and discussed in Chapter 8.

3.5. Conclusion

This chapter discussed information system research paradigms and research methodology candidates for carrying out this research. The Design Science Research (DSR) Methodology was selected to carry out this research. DSR methodology is appropriate and more consistent with the purpose of our research as it enables the researcher to understand the research problem and change the situation pertaining to control system security engineering by providing a supported solution and using associated methods to evaluate the feasibility of the solution. In Section 3.2.4, the selection of the DSR methodology was discussed and justified in relation to this particular research characteristics. The chapter illustrates the main phases of DSR methodology and maps our research into the DSR process, resulting in the five work packages presented throughout this thesis.

Next chapter presents a qualitative study assessing control system developers' needs in terms of security training and support.

Chapter 4

A Qualitative Study of Control System Developers' Support Needs for Security Engineering

Chapter Objectives

- To identify the current level of security awareness and knowledge of control engineers
- To enrich the understanding of ICS security engineering
- To identify the needs of developers in designing secure control systems
- To collect recommendations for the proposed framework

4.1- Introduction

This chapter presents a qualitative study that was carried out to enrich the understanding of the research problem and to capture the needs of developers in designing secure control systems. Research interviews were conducted with developers to explore the key issues of developing ICS security by design by synthesising the analysis of collected data in relation to the findings of the systematic literature review carried out in Chapter2.

As based on NIST guidance for building security training programs, it is essential to conduct a needs assessment before designing the training in order to allocate appropriate resources and techniques to meet the identified training needs. NIST suggested a number of assessment techniques, including reviewing current related trends published in the academic, government, or industry literature, and conducting interviews with key trainees (Wilson and Hash, 2003). Therefore, This study was conducted to explore the current level of developers' security knowledge and the needs of ICS developers in order to understand their support needs regarding security engineering.

Initially, the aim of the interviews is discussed as a part of the research objectives presented in section 4.2. The design of interview process is then explained and

justified, including participants' backgrounds and interview questions. Section 4.4, where the researcher discusses the candidate methods for analysing the collected data, justifies the selection of Thematic Analysis approach and explains the analysis procedure. Finally, the interview findings are presented and discussed in relation to the results of the systematic literature review and the aims of this research.

4.2- Aim of Interview

The main purpose to conducting the interviews is to gain an understanding of the issues arising when designing secure control systems, with respect to the role of system developers, by exploring the key factors of ICS security by design and discovering the existing support methods. The collected data is expected to attain the following goals:

Understanding the research problem: to gain further insight into the research problem identified earlier in Chapter 2 – that is, industrial control systems lacking security engineering by grasping the real situation relating to the design of control systems that is a part of the 'awareness of the problem' process in the DSR methodology, as discussed in Chapter 3, Section 3.4.1.2. In the context of the previous work, a number of interviews were conducted with domain experts to understand findings of the systematic review studied in Chapter 2. The interviews are intended to improve the knowledge gained from the literature in order to significantly enhance the understanding of the research problem and highlight the key issues required to find a solution by investigating the status of ICS development, current levels of security knowledge amongst ICS engineers, their security awareness regarding system design, and existing security support methods.

Needs assessment: by understanding ICS developers' needs and gathering their requirements. The collected data will help the researcher to identify the requirements of system developers in order that they pay increased attention to security during system design. The interviews are intended to identify developers' needs when designing secure systems. They also investigate the motivation to improve the security knowledge that can help gain solutions to the problem defined by both the systematic review, as presented in Chapter 2, and the interviews discussed in this chapter.

The qualitative data will be collected and analysed to partially answer the research questions introduced in Chapter 1 (Section 1.4) in order to contribute to the body of knowledge. The findings will illustrate the attitude of the developers toward applying security, the current obstacles to applying ICS security by design, and possible recommendations to overcome these obstacles.

4.3 Design of Interview Process

As this study involved human participants, it was essential to secure an ethical approval application before commencing data collection to ensure that it adheres to British Psychological Society (BPS) ethical guidelines. The ethical approval for the experiment was given by the Faculty Research Ethics Committee (FREC) (ref:1213/185) (Appendix B-1). It covered the issues related to respect participants; confidentiality of collected data and identity of participants; standard of self-determination, so participants can withdraw partially or completely from the interview; and honest and accurate representation of collected data.

The researcher applied the following guidelines prior to conducting the research interviews, as shown in Table 4.1:

- Ethics approval was obtained from the Faculty's Human Research Ethics Committee (see Appendix B).
- The time, date and place of the interviews were arranged with the participants.
- The respondents were shown an official letter obtained from the university declaring that data to be collected was for academic purposes.
- The aims of the research were conveyed to the respondents.
- A brief introduction about the research was given to the respondents.
- A digital recorder was used with the permission of respondents.
- A set of questions was used during the interviews (see Appendix C).

Table 4-1 Guidelines prior to starting the research interviews

The interview sessions were conducted in a one-to-one format with the aid of semi-structured interview questions, as presented in Appendix C.

The next sub-sections explain and justify the design of the research interview including interviewee selection, the size of the sample, and the interview research questions.

4.3.1- Surveyed Sample for Research Interview

This section presents the surveyed sample for the purpose of qualitative data collection. Interviewees were selected from a group of engineers pertinent to this research. The main criterion used in the selection process was that respondents be involved in the control system development process, and have previously worked in an industrial environment. This is to ensure that they have experience in developing control systems so that they can identify the issues of designing secure ICS within real development environments.

It is necessary, in order to conduct the research interviews, to estimate the sample size prior to data collection (Guest et al., 2006). Thus, related literature and guidelines for qualitative sample sizes were reviewed to ascertain a suitable sample size for an exploratory interview. Based on the survey of sampling size for qualitative research conducted by Guest et al., the majority of the literature recommended that sampling should continue until theoretical saturation occurs (Guest et al., 2006). Guest et al. carried out a study of sixty women, although their findings showed that data saturation had occurred as early as after their first six interviews.

Other researchers suggested guidelines for actual sample sizes that vary from one to hundreds. Baker et al. stated that a small sample, between six and a dozen, can be sufficient when a target population is a specific group or in some way hard to access (Baker et al., 2012). Morse suggested a guideline that recommends at least six participants during qualitative study (Morse, 1994). Creswell recommended between five to twenty-five participants (Creswell, 2012). Kuzel recommended that six to eight interviews should be sufficient for a homogeneous sample (Kuzel, 1992). Similarly, six interviews were conducted by Hanid in her study, and were felt to constitute a sufficient sample size (Hanid, 2014). Accordingly, studies involving a small sample are common in qualitative research (Crouch and McKenzie, 2006).

In an attempt to explore issues related to the design of secure control systems and identifying the needs of system developers, the researcher conducted seven interviews. The sample size was considered suitable for three reasons. First, the sample size is commonly used for a homogenous population, as described in the above literature. Secondly, the number of interviews covers all essential elements of the research questions. Thirdly, the study explores the current security awareness and knowledge among control engineers and the currently available support, while at the same time increasing the understanding of developers' needs and their attitudes toward security training support. These requirements provide the impetus to formulate better support for system developers in the design of secure systems. By doing so, the successful implementation of security support will meet developers' needs and help them to develop ICS security by design.

The researcher conducted seven interviews, two face-to-face, one by Skype interview, with the remaining four being phone interviews. The respondents were targeted based on their experience and their profiles on the university system and LinkedIn network. The individuals making up the sample worked in control system development in different countries including Libya, United Kingdom, India, Malaysia, Saudi Arabia and Belgium. Currently, two of the seven participants are working as lecturers, whilst the remaining five are Ph.D. students doing research on control systems engineering.

Every respondent was given a unique reference for the purpose of anonymity and fulfilling the requirements of ethical approval, as shown in Table 4.2. References were used during the data analysis process for better following and understanding, as presented in Section 4.4.

Participant Reference	Previous work/ experience	Organisation	Current wok	Interview Method
P1	System developer	De Montfort University, UK	Lecturer	Face-to-face
P2	Control Engineer	De Montfort University, UK	PhD student	Phone call
P3	System Designer	Ku Leuven, Belgium	PhD student	Skype call
P4	System Designer	De Montfort university, UK	Lecturer	Face-to-face
P5	Control Engineer	De Montfort University, UK	PhD student	Phone call
P6	Control Engineer	De Montfort University, UK	PhD student	Phone call
P7	System designer	De Montfort University, UK	PhD student	Phone call

Table 4-2 Participants' information

4.3.2- Interview Questions

The interview questions were designed as based on the aims of this study, which is that of enriching the understanding of the research problem and identifying developers' needs in terms of designing secure control systems (see Section 4.2). The study objectives were derived from fundamental issues revealed by the literature review presented in Chapter 2. The questions were grouped into three main sections (see Appendix C 1) and applied in the interview for data collection.

"Current security awareness and knowledge."

The interviewees were asked about control system security concerning the development process, including several probe sub-questions, as shown in Table 4.3. The questions were developed to explore awareness regarding ICS security issues, security by design, and the importance of security patterns and guidelines. They also

investigated the interviewees' current security knowledge by providing an example of the control system and asking the participants to identify where security could be applied to enhance the security level of a given system.

"Current support for developing secure ICS."

The participants were asked about the technologies used in system design, security-based technical assistance, and security training. The information gathered in this section will be used to explore the real-world circumstances of the development process and identifies the currently available support for designing secure systems.

"Developers' needs and requirements for security by design."

The proposal of developing a security training support mechanism was introduced and explained to the participants. The participants then were invited to identify any required support and desirable features. The information collected from this section identifies developers' needs, which were transferred to the form of IT requirements for the proposed support method.

Table 4.3 provides the justification for the interview research questions. The table explains the reasons for the proposed questions, referring to the research questions and the answers expected from the respondents.

Two documents were used in the interview. The first was a consent form, which the participants were required to sign before starting the conversation, whilst the second included the interview questions that were used by the researcher as a guide - see Appendix C. The interviews took place between 11:00 a.m. and 5:00 p.m. on working days. Each interview took about half an hour.

All participants permitted the researcher to use a voice recorder during interviews from which transcripts were produced so as to be able to apply an analytical procedure to the data collected, as discussed in the next section.

INTERVIEW QUESTIONS	INTERVIEW PROMPT	RELATED RESEARCH QUESTION
1- What are the security issues related to the Industrial Control Systems field?	To reveal awareness of ICS security issues	Part of research question RQ 1
2- In your opinion, at which phase of development cycle should security concerns become involved?	To reveal the awareness of the importance of security by design	Part of research question RQ 1
3- How do you determine whether your system design is secure?	To reveal the awareness of security guidelines and recommendations	Part of research question RQ 1
4- From your perspective, what are the most important security rules that developers should follow in order to design secure systems?	To reveal current levels of security skills	Part of research question RQ 1,2
5- Example discussion: if you design this system, where will you consider security policies?	To reveal the current level of security knowledge related to ICS security by design	Part of research question RQ1,2
6- Do you know secure design patterns or guidelines?	To reveal the awareness of security design patterns	Part of research question RQ1
7- How do you select the security patterns?	To reveal the current methods that can be used in selecting security patterns	Part of research question RQ1,2
8- What do you use for modelling? Does that tool support system security?	To reveal the currently available support related to design secure ICS	Part of research question RQ1,2
9- Have you attended any security training program?	To reveal the available security training programs and approaches	Part of research question RQ1,2
10- What kind of support could improve security knowledge of control engineers?	To reveal developers' needs and requirements	Part of research question RQ 2
11- What are features that would make a training tool more useful for engineers?	To reveal developers' requirements that would help in designing the training tool	Part of research question RQ 2
12- What are features of training tools that distract from learning?	To reveal undesirable features to be avoided when designing the training tool	Part of research question RQ 2

Table 4-3 Justification of interview questions

4.4- Data analysis

This section discusses the analytic methods applied to the data collected during the interviews. First, the qualitative data analysis approaches are reviewed, followed by the selected analysis approach, as presented and justified in Section 4.4.1. Then, the analysis process is explained in Section 4.4.2.

Having an interview plan and design, it was then appropriate to consider selecting a suitable approach for data analysis. The approaches commonly used in qualitative data analysis are thematic analysis and content analysis (Vaismoradi et al., 2013). Vaismoradi *et al*. conducted a comparison study to discuss the boundaries between the two approaches (Vaismoradi et al., 2013). Their findings showed that qualitative thematic analysis and content analysis have many similarities; however, the main difference is that content analysis aims to quantify content in a systematic manner. The above study concluded that both approaches can answer the same set of research questions, and they are robust enough to be used to conduct a research study. However, the authors believe that the quality of the data analysis depends on the effort spent by a researcher on the process of data collection and analysis, as well as the resulting interpretation and synthesis (Vaismoradi et al., 2013).

Although both approaches can be applied to the data collected in this research, thematic analysis approach was selected for the following reasons. First, it is the most widely used method to analyse interviews (Jugder, 2014). Secondly, "*rigorous thematic approach can produce an insightful analysis that answers particular research questions*" (Braun and Clarke, 2006). Also, this method complemented the purpose of this research by investigating interview data from two perspectives: the data perspective, which is driven through the coding process, and the research question perspective, whereby the data was checked to determine if it was consistent and answered the research questions. Finally, thematic analysis is an accessible and theoretically-flexible approach that can be used across a range of research questions and epistemologies (Braun and Clarke, 2006).

4.4.1 Thematic Analysis Approach

The qualitative thematic analysis approach was used to analyse the data collected in research interviews. Braun *et al*. described thematic analysis as "*a method for*

identifying, analyzing and reporting patterns (themes) within data" (Braun and Clarke, 2006). Thematic analysis can be based on prior categories, such as pre-figured or objective, or on categories that only emerge as the analysis proceeds (Braun and Clarke, 2006).

In the literature, a number of procedures were suggested to guide the thematic analysis of qualitative data. Creswell, and Miles and Huberman, recommended a three-stage analysis procedure, as follows: (Creswell, 2012) (Miles and Huberman, 1994)

1- Preparing the data for analysis by transcribing,
2- Reducing the data into themes through a process of coding, and
3- Representing the data.

Braun and Clarke stated that themes are identified through a rigorous process of data analysis, including the following main steps:

1- Data familiarisation,
2- Data coding, and
3- Theme development and revision.

Braun and Clarke also suggested splitting the above stages further into six phases, including familiarising with data, generating initial codes, searching for themes, reviewing themes, defining and naming themes, and producing the report (Braun and Clarke, 2006).

The next section explains the application of the thematic analysis approach to the qualitative data collected in the interviews.

4.4.2 Analysis procedure

The data were gathered through interviews with control system developers. Data analysis was guided by thematic analysis procedures as discussed in the previous section. The analysis process starts early in the data collection when the researcher begins to notice patterns within issues of interest in the data, and is ended by the reporting of the results of the analysis (Braun and Clarke, 2006).

Initially, audio records of the seven interviews were directly transcribed, as all interviews were conducted in the English language. Records were listened to many times to obtain data familiarisation and ensure the accuracy of the transcription.

Second, the coding process, where the transcripts were coded to capture important themes that represented patterned responses within the dataset. In thematic analysis, themes or patterns can be identified either in an inductive way or in a deductive (theoretical) way (Braun and Clarke, 2006). Inductive analysis, namely data-driven, is the process of coding the data without the researcher's assumptions or a pre-existed code frame, which means the themes are entirely linked to the data themselves (Patton, 1990). In contrast, deductive or theoretical analysis, namely analyst-driven, tends to be driven by the researcher's direct interests (Braun and Clarke, 2006). This means the analyst pays most attention to themes that have already been identified in previous research. This form of analysis provides greater detail as to certain aspects of the data in relation to the research questions. The choice between the two coding approaches maps onto the purpose of this study. Therefore, the researcher decided to choose deductive analysis to carry out the coding process.

The transcripts were thoroughly read and reread, and the data examined line by line to identify important codes by labelling relevant pieces that aligned with the research questions, as presented in the next section.

The coding process was performed manually, instead of using qualitative data analysis software, for a number of reasons: first, using analysis software is recommended when the sample size is large and can consume a considerable amount of the researcher's time. However, in small-scale samples, as in this study of seven interviews, manual analysis is recommended (Saldaña, 2015). Secondly, using software packages in the data analysis has no advantage over manual analysis (Halkier and Jensen, 2011). Furthermore, the most significant consideration for this research is that manual coding enriches the researcher's understanding and familiarity with the data (Scott, 2013).

At the third stage, that of theme development and revision, the codes were read, revised and aggregated to identify significant or recurring patterns that produced potential themes, as discussed in the results presented in the next section. First, the number of codes were reduced by extracting only the most important codes directly related to the study. Then, correlated codes were grouped under themes that were further reduced to conceptualising and generalising the data, as stated in reference (Rossman and Marshall, 1995).

The next section presents and explains the results of the analysis procedure applied to the qualitative data collected in interviews.

4.5- Results

This section presents the result of qualitative data analysis demonstrated in the previous section. Interview data were analysed with an emphasis on the purpose of this study. The preliminary results returned 108 codes; see Appendix C-2, that were further aggregated into twelve themes, as shown in Figure 4.1. The themes were further aggregated into main five themes: ICS design lack security, ICS Developers have some security awareness, ICS Developers lack security knowledge, ICS Developers lack support for design secure ICS, ICS Developers' requirements to design secure ICS, as shown in Figure 4.1.

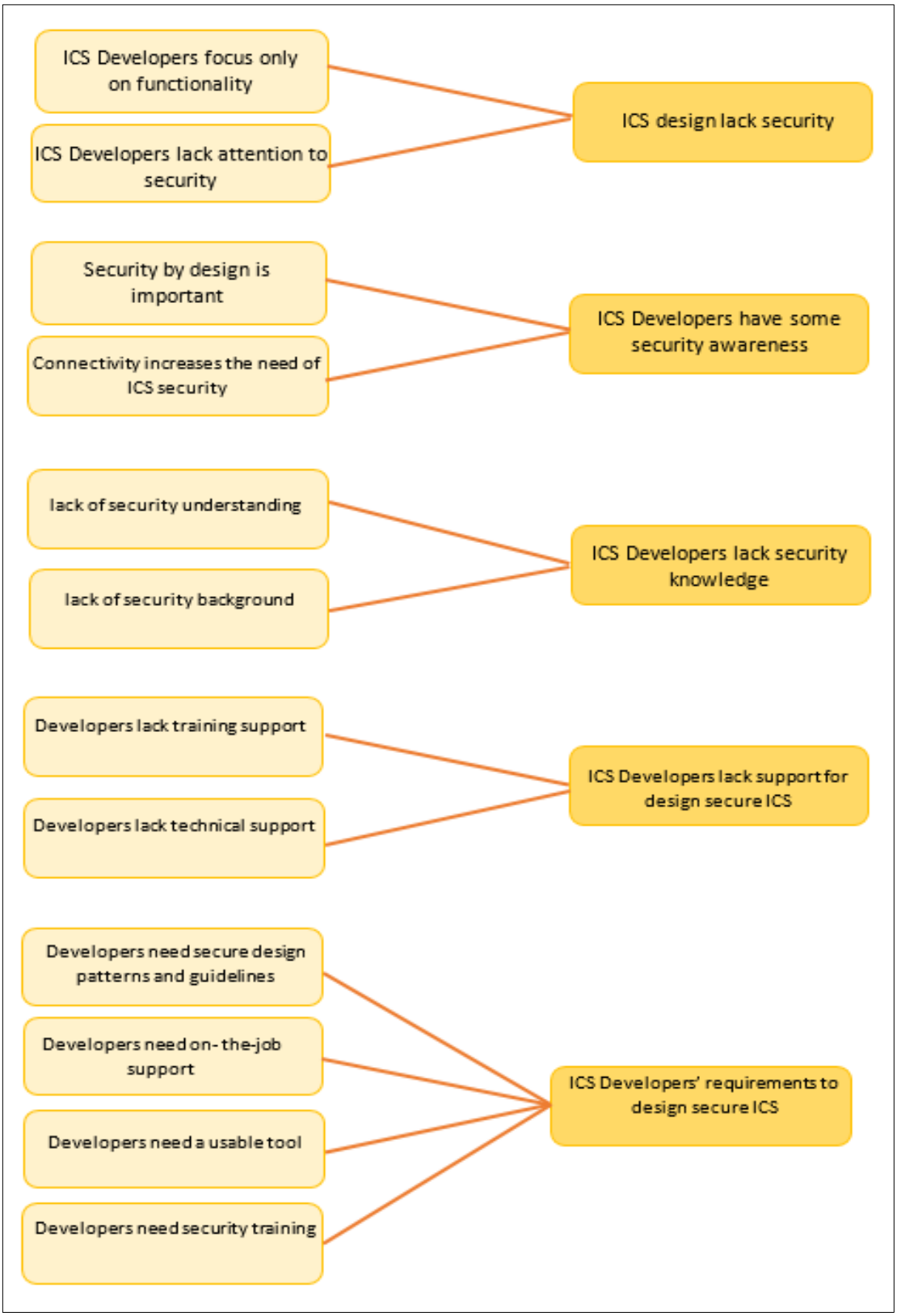


Figure 4-1 Themes map, developed from thematic analysis of interviews

The final analysis step, as recommended by reference (Braun and Clarke, 2006), is that of extracting related examples from the transcripts for developed themes, and relating these themes to the research questions. Table 4.4 shows the themes in relation to the topics investigated in the interviews, with some demonstrative examples of responses.

The results illustrate that ICS are developed without security considerations, as stated by [P5] "*we focus on controlling, we don't think about security…*", another participant [P4] said "*a lot of control systems are not secure, they can be modified, there is no access control…, systems tested just for functionality and performance*". However, responses show some awareness of the importance of security, for example: *"If the system is connected with other systems, I think you need security. As long as I have this connectivity, yes, should have security features…*" by [P4], "*because of industrial Internet of things, indutrie 4.0, and the interaction with cloud, I think all phases should have security*" by [P3].

The responses also show that ICS developers lack security knowledge; for example, "*I do believe that system should be secured, but my background is far away from security…*" by [P2], another respondent said "*we never been taught how to protect our system*" [P7].

The respondents claim that they lack support, as stated by [P4] "*none of the tools allow you to do security…*"; [P2] said "*The biggest issue in ICS security is the training, we don't do regular training, or workshops…*". They stated that they needed training support: "*We need support thorough learning, we need to understand security, understand the principles…*" by [P1], a usable tool, "*we need a tool that gives suggestions for security, suggest some patterns to choose from,…*" by [P3], security guidelines, "*we need guidelines to follow, we need knowledge given by security engineer, to know what kind of security is appropriate…*" [P7], and that they need support during system development, as [P7] said "*we need security solutions according to our needs, I mean system, and corresponding learning material to understand weaknesses in our system,…*" Similarly, [P3] said "*the tools we use don't allow us to include security or to model security property…*"

Topics	Examples of the responses	Themes
Current security awareness and knowledge	"In the past, security wasn't critical thing... but now the Internet increases the importance of security" [p1] "After using the Internet, there are security issues that carry risks…, systems should be secured from scratch…" [p2] "Security wasn't a problem, but now it is... because of industrial Internet of things, indutrie 4.0, and the interaction with cloud, I think all phases should have security" [p3] "If the system is connected with other systems, I think you need security. As long as I have this connectivity, yes, should have security features…" [p4] "It has to be done properly, of course, at the beginning of the design stage" [p4] "It is better to implement security from the beginning" [p5] "Security is important at the structural and operational level" [p7]	ICS Developers have some security awareness
	"we care about functionality more than security…" [P1] "we don't focus on security…" [P2] "we deliver systems without security…" [P3] "a lot of control systems are not secure, they can be modified, there is no access control…, systems tested just for functionality and performance" [p4] "we focus on controlling, we don't think about security…" [p5] "none of the tool allow you to do security…" [P4] "we rely on the vendor or third party for security, we choose a good vendor" [P6] "we can't do much about security, we don't have the flexibility to do security…" [P7]	ICS design lack security
	"I don't know about security…." [P1] "I do believe that system should be secured, but my background is far away from security…" [P2] "The biggest issue in ICS security is the training, we don't do regular training, or workshops…" [P2] "We have 'zero' knowledge on security, we need training, workshops" [P3] "no security training, even in my education…" [P4] "I don't have good security background, I've done training course before, but it's very general…" [P5] "we never been taught how to protect our system…" [P7]	Developers lack security knowledge

Current support for developing secure ICS	"we don't do regular training, or workshops…" [P2] "the tools we use don't allow us to include security or to model security property…" [P3] "none of the tool allow to do security…" [P4] "Tools don't support security…, no security training…" [P5] "nothing supports you to do security, no understanding, and the tool I'm using doesn't allow me…" [P6] "There is no guideline, engineering methodologies don't include security, there is no common concept for security of any methodology…" [P7]	ICS Developers lack support for design secure ICS
Developers' needs and requirements for security by design	"we need support thorough learning, we need to understand security, understand the principles…" [P1] "we should connected with security experts, we should be working together or have regular meeting…" [P2] "we need a tool that gives suggestions for security, suggest some patterns to choose from…" [P3] "we need to understand security, we need something simple and efficient, security is complex, avoid complication…" [P4] "we need to know how similar problems solved, we need personal support…" [P5] "we need to understand weaknesses, we need training courses…" [P6] " we need guidelines to follow, we need knowledge given by security engineer, to know what kind of security is appropriate…" [P7] "we need security solutions according to our needs, I mean system, configurable solutions, and corresponding learning material to understand weaknesses in our system, in understandable language…" [P7] "we never been taught how to protect our system, or how to implement security…" [P7]	ICS Developers' requirements to design secure ICS

Table 4-4 Results of data analysis

Table 4.5 summarises the interviews' findings in relation to those revealed by the literature review. In the past, ICS security was not a concern where ICSs were isolated from external networks, and where the emphasis was on other requirements such as performance and functionality (Drias et al., 2015). However, since the ICSs began to be connected with the Internet by adopting IT technologies, security becomes a necessary requirement of developing ICS (Drias et al., 2015) (Durrani et al., 2013) (Fernandez et al., 2011). Although ICS developers had some awareness of

this necessity and showed a readiness to improve system security, they do not generally consider security in system design; rather, they focus only on functionality and safety, which ultimately leads to delivering ICSs with security weaknesses.

Systematic Literature Review Findings		Interview Findings	
Findings	Sources	Participants' responses	Sources (participant reference)
Security is becoming a real concern for both ICS vendors and owners	(Drias et al., 2015 (Durrani, 2013 #256) (Fernandez et al., 2011)	Internet connectivity increases the need of security	(P1, P2, P3, P4, P5, P7)
ICSs are not sufficiently secured and need more research effort	(Hadziosmanovic et al., 2012) (Krotofil and Gollmann, 2013)	ICSs are delivered without security	(P1, P2, P3, P4, P5, p6 P7)
The importance of ICS security by design	(Fernandez et al., 2011) (Zineddine, 2016) 2016) (Fernandez et al., 2008) (Oates, 2005) (Ur-Rehman and Zivic, 2015)	Security should be implemented from the beginning; at all development phases	(P1, P2, P3, P4, P5, P7)
ICS vendors and security professional must bridge the gap and work together for feasible solutions	(Yang and Zhao, 2014) (Kunsman et al., 2015) (Brundle and Naedele, 2008) (Zineddine, 2016)	The need to connect with security engineers and working together	(P2, P3, P4, P7)
ICS Security awareness and education is a real concern	(Amaechi and Counsell, 2012) (Boyes, 2015) (Miyachi and Yamada, 2014) (Durrani et al., 2013) (Vaughn Jr and Morris, 2016)	The training is most important in improving ICS security	(P4, P7)
Lack of security awareness and knowledge	(Amaechi and Counsell, 2012) (Savola and Ahonen, 2006) (Vaughn Jr and Morris, 2016) (Graham et al., 2016) (Axelrod, 2011) (Ismail et al., 2014)	Lack of security knowledge	(P1, P2, P3, P4, P5, P6, P7)
Recommendation for educate developers in	(Foo et al., 2013) (Nelso and Chaffin, 2011)	The need of security training support	(P1, P2, P3, P4, P5, P6, P7)

security and best practice			
The need of methods and tools that support ICS security engineering	(CPNI, 2016) (ICS-CERT) (Motii et al., 2015)	The need of security guidelines and methodologies	(P3, P7)

Table 4-5 The interview findings in relation to the findings of the literature review

4.6- Discussion

The data collected from the respondents' interviews are presented graphically, as presented in Figure 4.1 in Section 4.5, and narratively in the previous section. This section discusses and provides an interpretation of the findings through reviewing and summarising the study results in response to the research questions, and in reference to the previous literature that was systematically reviewed in Chapter 2.

The participants were asked about ICS security engineering regarding the need for security, development processes, security background, guidelines and standards, tool support, and training support. The information collected was to enrich the understanding of ICS security engineering within the ICS developers' community.

Based on the analysis given in the previous sections, this study revealed two key reasons for the lack of security consideration throughout the system development cycle. First, ICS developers do not have sufficient security knowledge to implement security in their systems. Ordinarily, they do not have any security training or education, as one of the respondents said "*We never been taught how to protect our system, or how to implement security…*" [P7]. The lack of security training was also outlined in the literature as one of the main problems associated with ICS security (Vaughn Jr. and Morris, 2016) (Graham et al., 2016).

The second key issue is a lack of support. The respondents claimed that they lack security support in both technical, i.e., within the system development environment, and training, support. Although the literature proposed several support methods, most of which are developed for risk assessment, which support the security of the system at the operational level, they do not focus on the learning aspect, mainly providing security awareness without training and education (Foo et al., 2013). The literature

also stressed the importance of the learning aspect of any proposed method (Annex, 2011).

To summarise, the results of this study clearly stressed the need of support within the system development process to increase the potential for obtaining security by design. The information collected in the interviews enriched the understanding of the problem and identified the key factors for improving ICS security at the structural level throughout the system development cycle. The study shows that ICS developers do not pay attention to the system security aspect. However, they are willing to enhance their current situation through various initiatives such as training support, security guidelines, and security-related techniques.

4.7- Conclusion

This chapter presented a qualitative research study to explore the problem of building insufficiently secured control systems. Interviews with ICS developers were conducted to gather their views in terms of security engineering and to identify their needs. The objectives of the study were met through the findings obtained from the interview analysis process. The results of the study were consistent with the results obtained from the systematic literature review presented in Chapter 2. The systematic review revealed a number of challenges in building secure control systems. However, conducting the interview study further supported the empirical evidence of the issues reported relating to ICS security. In addition, the results of the research interviews complemented the understanding of these problems and clearly indicated the current knowledge gap - which this research will contribute to by filling - by adding new insights into ICS security from the perspective of the participants in this study. The contribution of this study is that it is the first such attempt, to the best of our knowledge, to explore the needs of control system developers in enabling a security by design approach. The study revealed that control systems lack security engineering for two key reasons: first, control system developers lack security training; second, developers lack support for security engineering throughout the development cycle. The findings drew attention to the role of developers in building secure control systems and derived a solution in the form of an educationally supported security engineering framework, as proposed in this research.

The chapter discussed the main issues associated with the interview design, selected appropriate participants, and the sample size was justified on the basis of all the factors that might theoretically have helped the researcher to determine a suitable size for this study. In addition, the data analysis method was discussed and justified, followed by the study findings and discussion in relation to the results of the systematic literature review and the research questions.

The next chapter elaborates on the proposed framework for supporting ICS developers in designing secure control systems.

Chapter 5
Industrial Control System Security Engineering Support (ICS-SES) Framework

Chapter objectives

- To introduce the proposed method for supporting ICS developers in designing secure systems
- To demonstrate the proposed pattern-based security guide
- To demonstrate the proposed embedded training method
- To present the ICS-SES framework for supporting ICS security engineering
- To illustrate ICS-SES architecture
- To explain the workflow supporting the ICS security engineering process

5.1-Introduction

Chapter 2 and Chapter 4 showed that there is a need to support ICS developers to improve their security knowledge such that they are better able to build secure control systems. This chapter introduces a novel framework that is intended to support ICS developers in designing secure systems by bridging the associated knowledge gap and improving their security knowledge. The originality of our method lies in its assembly of two methods, namely Pattern-based Security Guide and Embedded Security Training, to assist system designers in improving their security skills and their understanding of security in general, and by consequence enabling ICS security by design. The support method is based on the adaption of security design patterns, a problem-based learning approach, on-the-job security training, tailored training, and technical innovations.

Our method focuses on supporting ICS developers in two main dimensions. First, it provides technical support by guiding developers as to the selection of a suitable security design pattern to be configured in a system model as based on the required security property required to mitigate a security flaw that has become apparent. Second, it provides a knowledge dimension by suggesting personalised learning material related to the security problem, possible risks and nominated solution

patterns. Figure 5.1 demonstrates the proposed method in relation to the research problem being addressed and desirable goals.

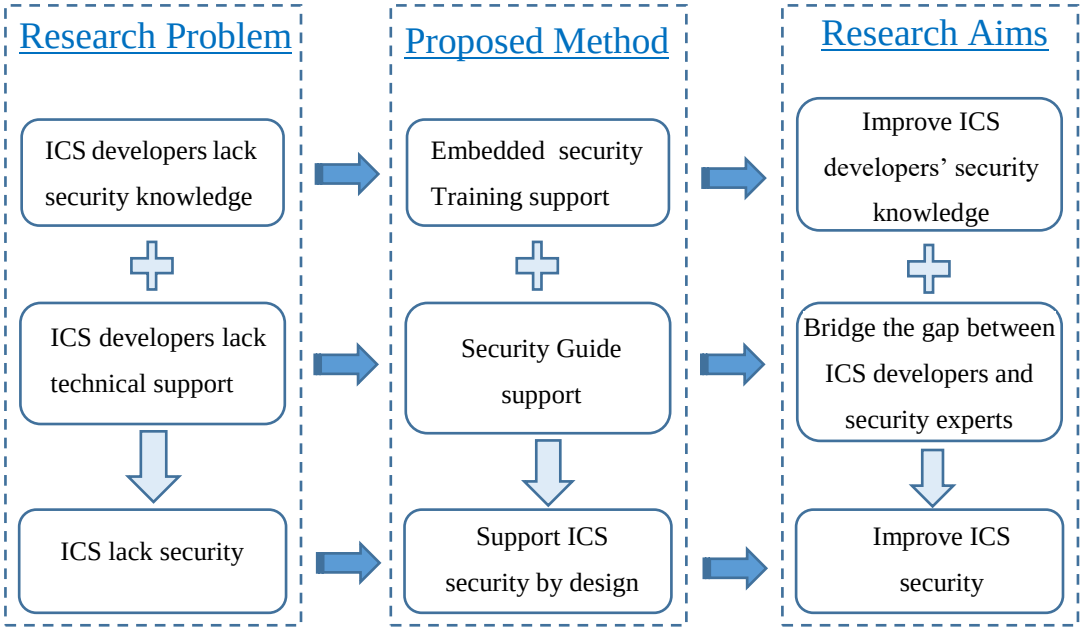


Figure 5-1 Proposed method in relation to the research problem and research aims

This chapter is organised into the following sections: Section 5.2 introduces the ICS Security Engineering Support (ICS-SES) framework that integrates two methods of a 'Pattern-Based Security Guide', which outlines the process of pattern selection, and 'Embedded Security Training', which demonstrates the training process, and explains the entire support process. Section 5.3 outlines the system requirements for the ICS-SES tool. Section 5.4 introduces the ICS-SES architecture and discusses the development process. Section 5.5 demonstrates the ICS-SES workflow and explains the process of supporting security by design using security guides and security training. Section 5.6 summarises this chapter.

5.2- Industrial Control System Security Engineering Support (ICS-SES) Framework

5.2.1- The Rationale behind the ICS-SES Framework

This section provides insight into the approaches that motivated our proposed support method and the techniques that were used in the ICS-SES framework.

The Problem-Based Learning (PBL) method was adopted in our supported framework by providing security guidance and training based on a deliberate security flaw produced by an engineer during system design. PBL has been defined by Barrows and Tamblyn as "*the basic human learning process that allowed primitive man to survive in his environment,…, it is the learning that results from the process of working toward the understanding or resolution of a problem."* (Barrows and Tamblyn, 1980). PBL strategy has been applied in engineering education for many years across a variety of professional engineering schools using numerous types of problems based on the nature of the discipline (Jonassen and Hung, 2008, Mills and Treagust, 2003). For example, PBL has been applied in chemical engineering (Woods, 1996), architecture (Donaldson, 1989, Maitland, 1991), and to solve design problems (Cawley, 1989). PBL offers good prospects for learning, especially with the aid of guided teaching and tutorials (Perrenet et al., 2000). By using our problem-based security learning, ICS developers can be more self-regulated and effectively transfer any skills attained into real-world scenarios and retain knowledge for a longer time than is generally associated with more traditional learning methods (Norman and Schmidt, 2016).

Our supported training method was designed as on-the-job-based learning, where security learning is embedded in everyday work. On-the-job training (OTJ) is one of the Higher Education industries' methods for developing required competences within their graduates by the transferral of skills into working experience (Bernardo et al., 2014); indeed, most of the learning occurs in the work setting itself (Jacobs, 2003). Training has been defined by NIST as follows "*The 'Training' level of the learning continuum strives to produce relevant and needed security skills and competencies by practitioners of functional specialties other than IT security (e.g., management, systems design and development, acquisition, auditing).*" NIST Special publication-800 (Wilson and Hash, 2003).

OTJ training is commonly acknowledged as being a useful method, such as learning by solving problems in the workplace (Boud and Rooney, 2015). The use of well-planned OTJ training enables workers to effectively learn new skills that meet organisational needs (Rothwell and Kazanas, 2004). It also leads to high-quality service as employees continuously improve the quality of the product, and they usually prefer OTJ more than off-the-job training (Rothwell and Kazanas, 2004). In

addition, our embedded security training targets ICS engineers who typically learn through practise by ‘doing’ or observing in the workplace (Rooney et al., 2014). Therefore, OTJ security training can significantly improve the performance of ICS developers in terms of designing secure systems (Saks and Burke-Smalley, 2014).

Using the on-the-job training method requires a technique that is capable of delivering tailored training to be employed. An automated planner was used in order to provide personalised support to a system engineer.

In the learning context, personalisation is also referred to as individualized learning (Sebba et al., 2007). However, using the term ‘individualised learning’ is a bit unrealistic and places more pressure to provide the exact materials required for each individual learner (Johnson, 2004). By contrast, ‘personalised learning’ is more suitable as it can refer to learning in a small group or even down to being on a one-to-one basis (Sebba et al., 2007). Therefore, the term ‘personalised training’ is used in this thesis rather than ‘individualised training’.

Since offering personalised learning programs can promote better learning (Garrido et al., 2011), our security training method was designed to meet the personal needs of ICS developers through the adoption of an automated planning technique.

5.2.2- Pattern-Based Security Guide

The Security Guide proposed in this research is based on the use of security patterns to assist designers in securing their systems. Security patterns (see review, Chapter 2, Section 2.2.7) were used to bridge the cultural gap between security experts and ICS developers by capturing security expertise in the form of security patterns.

Researchers have already attempted to integrate security patterns into system development cycles in software engineering (Fernandez-Buglioni, 2013) (Maña et al., 2013) (Arjona et al., 2014) (Nguyen, 2015) (Hamid et al., 2016). However, to the best of our knowledge, this is the first attempt to provide a practical pattern-based security guide that employs security patterns and supports their selection in ICS product line engineering, particularly at the system design phase.

Our security guide initially takes the result of an external security scanner as its input, including a security problem and a vulnerable asset. Then, it guides system designers to select appropriate security design patterns to allow them to solve the associated security problems.

The following subsections discuss the development of the security patterns catalogue and explain the pattern selection process.

5.2.2.1- Vulnerability-based Security Patterns Catalogue

Security patterns have been classified in a number of studies with respect to various dimensions (Mouratidis, 2006) (Nelso and Chaffin, 2011) (Schumacher et al., 2013) (Motii et al., 2015) (ICS-CERT). However, none of these studies proposed a systematically means of directly mapping between system vulnerabilities and security patterns. As discussed in the above sections, this security guide is intended to guide an engineer in the selection of applicable security solutions, here presented as security design patterns, based on an indentified vulnerability. Therefore, it was necessary to create a catalogue that systematically relates security patterns with system design security flaws in relation to vulnerable assets, potential risks and security requirements. The catalogue was created by adapting the ICS-CERT category (Nelso and Chaffin, 2011) and Motii *et al.*'s classification (Motii et al., 2015), as shown in Figure 5.2. ICS-CERT published common ICS vulnerabilities with a list of related security threats, while in Motii *et al*. classification, security patterns were guided by a security risk assessment by identifying security requirements and relating them to security patterns based on Non-Functional Requirements (NFR), i.e., the NFR-based approach proposed by Weiss and Mouratidis (Weiss and Mouratidis, 2008).

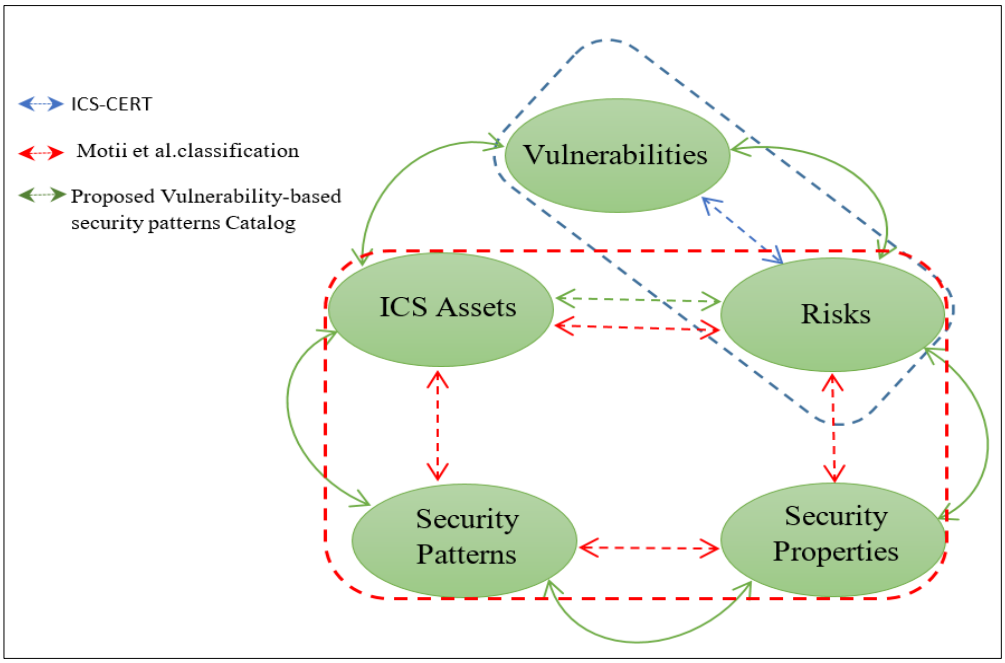


Figure 5-2 Vulnerability-based security patterns Catalogue

(based on references (Nelso and Chaffin, 2011) (Motii et al., 2015))

Both classification systems were thoroughly reviewed and linked where necessary. The catalogue was developed by relating each vulnerability to a set of applicable security patterns that satisfy the security requirements derived from the potential risks associated with a particular asset's vulnerabilities.

ICS Assets were categorised into four main categories based on ICS-CERT: centre controller (e.g., HMI, SCADA server), field controller (e.g., PLC, RTU, IED) and field units (e.g., actuators, meters, sensors) and network communication.

ICS-CERT categorisation of vulnerabilities was also adopted, including those of improper input validation, improper authentication, improper access control, lack of audit and accountability, lack of backup facilities, unencrypted sensitive data, and improper software configuration and management.

Security risks were classified based on the work of Masood (Masood, 2016) that used Microsoft's STRIDE model (Howard and LeBlanc, 2003), namely **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, and **E**levation of Privilege. All risks identified in reference (Motii et al., 2015) were categorized into the STRIDE model by relating vulnerabilities and assets.

The generic security requirements for ICS include mutual authentication, confidentiality, authorisation, data integrity, non-repudiation, system security capability monitoring, and audit and availability (Masood, 2016).

Concrete security design patterns were categorised into a set of abstract security patterns based on Fernandez *et al*. including authentication, authorization, security logger and auditor (Fernandez-Buglioni, 2013). Abstract security patterns were assigned to security requirements as suggested by Motii et al. (Motii et al., 2015)

The catalogue was created to guide and enable the automated process of pattern selection, so the tool can automatically derive a number of security pattern candidates with regards to a security problem that has become apparent, as explained in the next section.

5.2.2.2- Security Pattern Selection

Selecting an appropriate security pattern plays an important role in pattern-based secure system engineering methodology. Over the last decade, there has been considerable effort expended in undertaking this subject. Weiss and Mouratidis proposed a pattern selection method by formalising a security pattern in Goal-oriented Requirements Language (GRL) based on security properties and threats (Weiss and Mouratidis, 2008). Hasheminejad and Jalili used a text processing approach and learning techniques in their proposed method for automatic pattern selection (Hasheminejad and Jalili, 2009). Fernandez *et al*. presented a pattern-based development methodology with respect to multidimensional pattern classification according to the development phase (Fernandez et al., 2011). A classification was proposed that relies on the application domain, pattern recognition needs and security properties (e.g., confidentiality, integrity, accountability, availability, authorisation and authentication) (Bunke et al., 2012).

The intention behind our pattern selection process is to find a set of security design pattern candidates that satisfy the security requirements identified as based on a security vulnerability in a system model. The selection method is guided by the security pattern catalogue developed, as presented in the previous section. It follows four main steps: first, security risks are identified based on a given security vulnerability and ICS asset. Second, identifying security requirements by responding

to security risks associated with a vulnerability. Third, abstract security patterns are selected to satisfy the security requirements issued. Lastly, related concrete security patterns are identified and provided to a system designer to be integrated and evaluated.

For example, the selection process for a vulnerable system network that has an unencrypted sensitive data flaw, which has been detected by an external security analyst, is as following:

Step 1: identifying security risks associated with the detected vulnerability, 'unencrypted sensitive data', which are spoofing, tampering and information disclosure, and the ICS asset 'network'.

Step 2: according to the security risks identified in Step 1, three security requirements are identified, namely 'confidentiality of data, integrity of data, and mutual authentication'.

Step 3: abstract security pattern, Virtual Private Network (VPN), was selected to satisfy the security requirements issued in Step 2 that are applicable to the ICS asset.

Step 4: VPN security pattern has two concrete security patterns: IPsec VPN and TLS VPN.

The method refines the selection process further by making use of a pattern application history that stores successful pattern applications in a certain design context. Each time a pattern is selected and evaluated, a designer gives feedback that can help to improve the precision of the following selection process for the same problem context.

The security guide is intended to assist engineers in pattern selection and in facilitating the identification of training objectives in the second supported method, 'embedded security training'.

5.2.3- Embedded Security Training

In having a set of suggested security patterns that can potentially solve a design security problem, it is beneficial to provide learning material about the security

problem and educate system engineers regarding the security patterns. Our training method is intended to offer contextualised and personalised security training to improve the security knowledge of ICS developers. The training method takes a security pattern candidate, which is produced by the Security Guide, as input and provides personalised learning material by applying an automated planning technique.

Effective learning was defined by Litzinger *et al.* as "*those that support the development of deep understanding organized around key concepts and general principles, the development of skills, both technical and professional, and the application of knowledge and skills to problems that are representative of those faced by practicing engineers*." (Litzinger et al., 2011).

The Embedded Security Training method was designed in line with the ADDIE model that comprises five basic steps: Analysis, Design, Development, Implementation, and Evaluation (Molenda, 2003). The model originated from Instructional Systems Development (ISD) (Wilson and Hash, 2003). ADDIE is a common and effective model used by training developers and instructional designers (Kovalchick and Dawson, 2004). The steps taken in ADDIE are recursive, as shown in Figure 5.3. The five basic phases of the model are Analysis, Design, Development, Implementation and Evaluation.

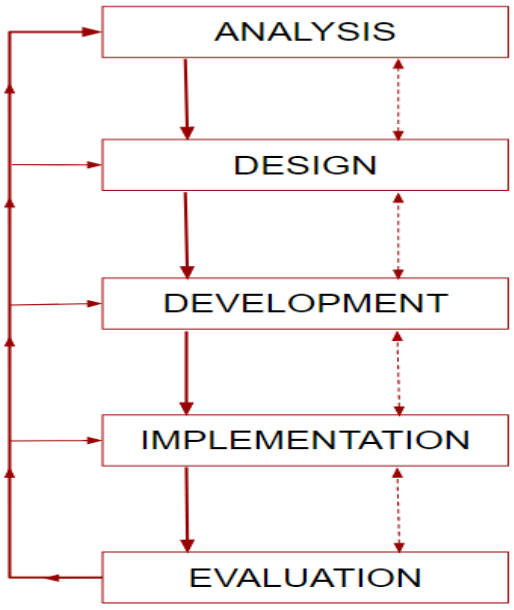


Figure 5-3 ADDIE processes

(source: reference (Molenda, 2003))

Table 5.1 illustrates the adaption of the ADDIE model into our security training method. It includes three main phases, namely training needs analysis, training plan design and training plan execution.

ADDIE Model	Proposed Training Method	Description
Analysis	Training Needs Analysis Knowledge Engineering	Creating a planning domain and planning problem based on a learner's background and training needs
Design and Development	Training plan design	AI planner generates a tailored training plan by making use of a planning problem and planning domain
Implementation and Evaluation	Training plan execution	Delivering training material to a trainee

Table 5-1 Security training process adapted from ADDIE model

(source: reference (Molenda, 2003))

1) ***Training Needs Analysis*** - in this phase, the previous knowledge is analysed to identify what is to be learnt. This task is performed through knowledge engineering.
2) ***Training Plan Design*** - a phase where a personalised training plan is generated to meet the trainees' needs, as defined in the previous phase.
3) ***Training Plan Execution*** – in this phase, training material is delivered to trainees, where they start to perform a sequence of learning activities.

5.2.3.1- Training Material

According to Polsani, *"as individual words cannot independently produce meaning, the LOs in themselves are insufficient to generate significant instruction [. . .] How many LOs, how they are related, and for what purposes will be determined by the instructor's objectives, pedagogical methodology and instructional design theories."* (Polsani, 2006). This indicates the necessity of using learning objects and their interrelations to define a learning program. Therefore, the initial step to the proposed security training method is to define security training objects by designing or reusing the available learning repositories, such as the ICS-CERT training material.

Training material is the learning sources that are constituted of a set of learning objects. Learning objects have been defined by Mavrommatis as "*A Learning Object is a standalone, reusable, digital resource that aims at teaching one or more instructional objectives or concepts"* (Mavrommatis, 2008).

In this work, a set of security training topics were defined with their relations and dependencies in reference to security patterns. Metadata was also defined for learning objects for use in the knowledge engineering process. The information identified in the metadata set is part of the Instructional Management Systems (IMS) standards and specifications relevant to e-learning, and particularly to learning objects (Friesen, 2005). Five attributes were defined for each security learning object including the identifier, title, learning time, relations and learning outcomes. However, this information is not suitable for direct use by the AI planner and needs to be translated into a planning model, as explained in the next section.

5.2.3.2- Using AI automated planning in Embedded Security Training

AI planning was defined by Garrido *et al*. as the "*task of finding a solution within a search space*" (Garrido et al., 2011). A plan, for a given initial state, is sequence of activities that achieves a set of desirable goals (Camacho et al., 2008). An automated planner requires two particular files that are essential to its completion of the planning process, planning domain and planning problem. Figure 5.4 presents a high-level description of the inputs and outputs of a planner.

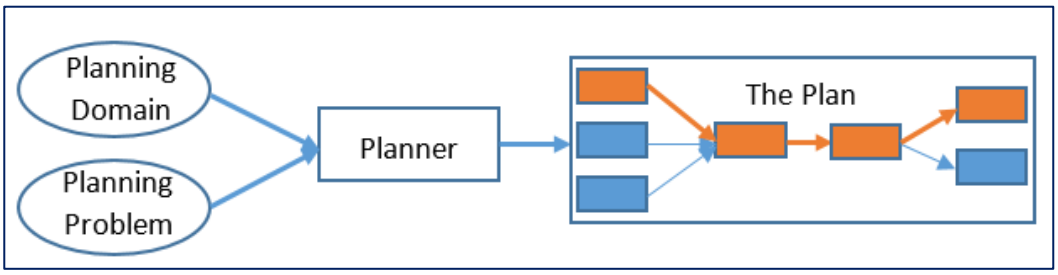


Figure 5-4 inputs and outputs of a planner

(Adopted from reference (Garrido et al., 2011))

Automated planning was first used for learning in Intelligent Tutoring Systems, as proposed in reference (Peachey and McCalla, 1986). AI automated planning techniques were successfully used to enhance the learning process and have resulted

in significant advances in e-learning (Onaindia et al., 2007). Garrido *et al.* also used an AI planner in their proposed approach, namely myPTutor, to personalise e-learning course design (Garrido et al., 2012). However, they are based on Case-Based Reasoning (CBR) in order to save predefined learning plans in a library, since learning plans for a course were recurrent, whereas our work is on-the-job-based learning and plans differ from one trainee to another.

The planning technique was used in the security training to provide personalised learning throughout an adaptive training process, where training materials are tailored to the context of a specific problem and personal training needs. Using an intelligent learning system can eliminate the subjectivity of knowledge assessment and raise its objectivity to a higher level (Kresimir et al., 2014). The training method benefits from the features of an automated planner, as follows:

- Planning domain is used to represent training objects and their relationships.
- Planning problem is used to define individual training cases.
- AI planner is used to generate a training plan.
- Preconditions and effects are used to represent prerequisites and learning outcomes, respectively.

A planning domain and planning problem were created through the knowledge engineering process to be used by an automated planner, as explained in the next subsections.

5.2.3.2.1- Knowledge Engineering

Once the metadata set of the security training material was defined, it was essential, in order to use the AI planning technique, to undertake a knowledge engineering task, namely a planning domain, by mapping training material into a planning model. A planning domain was created based on one of the approaches presented in reference (Garrido et al., 2009) that focusses on how to compile learning objects and student profiles into planning domains and problems. Table 5.2 illustrates how the training metadata was translated into a planning domain. All learning objects were mapped into actions and relations, and dependencies were compiled into preconditions. Learning outcomes were mapped into effects. Solution plans represent training plans

that consists of a sequence of learning objects that an engineer needs to perform to understand a certain topic.

Training metadata	Planning Domain
Learning object identifier	Action
Title of Learning object	Action name
Learning time	Duration
Relations and dependencies	Preconditions
Learning outcomes	Effects
Tailored training plans	Solution plans

Table 5-2 Mapping training metadata into the planning domain

(source: reference (Garrido et al., 2009))

The second task of the knowledge engineering process is the planning problem. The planning problem initialises the variables that represent objects, the initial state and the goals (Garrido et al., 2009). In the work presented in this thesis, the planning problem, which describes the training needs, is applied to the planning domain in order to generate an adaptive training plan that is tailored to a trainee's background and needs.

Table 5.3 illustrates mapping the individual training case into a planning problem. The object represents a trainee. The initial state represents the trainee's prior knowledge, which can be retrieved from the training history. The goal represents a security pattern selected by the Security Guide.

Training case	Planning Problem
Trainee	Object
Trainee's prior knowledge	Initial state
Security pattern	Goal

Table 5-3 Mapping a training case onto a planning problem

(source: reference (Garrido et al., 2009))

A planning problem is modeled with every training case, whereas a planning domain is modeled once for the metadata of a learning object and reused by a planner within all training cases.

5.2.3.2.2- Training Plan Execution

An automated planner uses the two modelled files, the planning domain and problem, to generate a personalised training plan that includes a sequence of learning objects. Once a training plan is produced, it is executed by retrieving the corresponding training material and displaying it to the trainee.

The two proposed methods for supporting developers in ICS security by design was integrated into the ICS-SES framework, as depicted in Figure 5.5. The framework was developed to support ICS developers in designing secure systems and in improving their security knowledge during the system design phase.

Since our framework provides problem-based support, an external security analyser was used to detect a security problem in a system model. The ICS-SES framework supports system designers within the work environment as follows:

- The Security Guide applies a selection method and identifies a set of suggested security design patterns that have the potential to solve the detected problem.
- The suggested patterns are provided to a designer to choose a solution pattern based on the designer's preference.
- Once a designer chooses a pattern, the metadata of that pattern will be sent to the Embedded Security Training to be translated, along with the background of a designer, into the planning problem.
- The AI planner applies the planning problem to the predefined planning domain and generates a training plan that guides the security training provided to an engineer.
- When a designer selects a pattern from the list of security pattern candidates, the pattern is evaluated by configuring it to a system model and using a security analyser.

Our supported method is viable from two perspectives. On the one hand, guiding ICS developers in selecting security design patterns can improve control system security. Providing contextualised solutions can be more effective than general guidelines and standards.

On the other hand, on-the-job security training can improve the knowledge of ICS developers, providing an adaptive training that is tailored to the design context and needs of an engineer. Using security patterns can bridge the gap between security experts and control system professionals. The advantages of our framework lie in the utilisation of both methods together as per 'Security Guide and Security Training'.

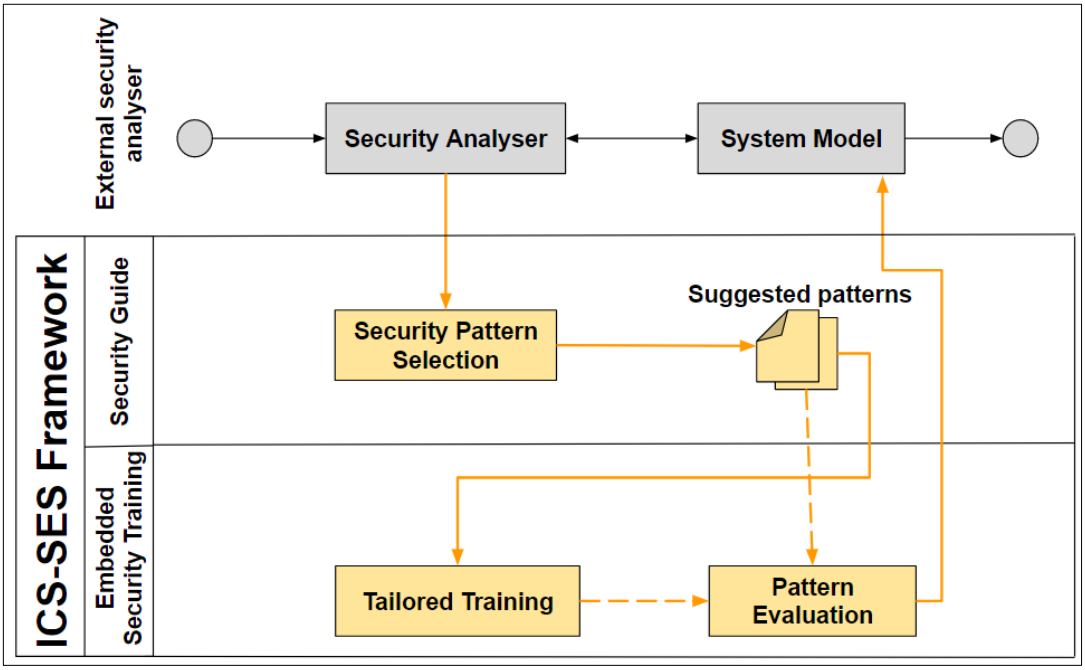


Figure 5-5 ICS-SES Framework, ICS security by design

5.3- System Requirements

ICS-SES framework is proposed as an on-the-job support tool to help engineers in designing secure systems. Since it was designed to be used in the work environment, it requires integration with a system modelling tool such as Papyrus, or Enterprise Architect (Sparx Systems). This integration allows a designer to configure a selected pattern to a system model that needs to be tested as to whether it improves system security.

ICS-SES also requires a security analyser to be used during the system design phase to identify security weaknesses and vulnerable assets that are used as input data to

ICS-SES. Using an analyser also plays an important role in evaluating a selected security pattern, where the security of a system model is analysed and evaluated after pattern configuration.

5.4- ICS-SES Architecture

This section presents the ICS-SES architecture and articulates the functionality of our method in supporting ICS security engineering.

Figure 5.6 shows the ICS-SES architecture. This work implemented the components colored orange, while green components, which present external tools and resources, were employed to complement the goals of the proposed method.

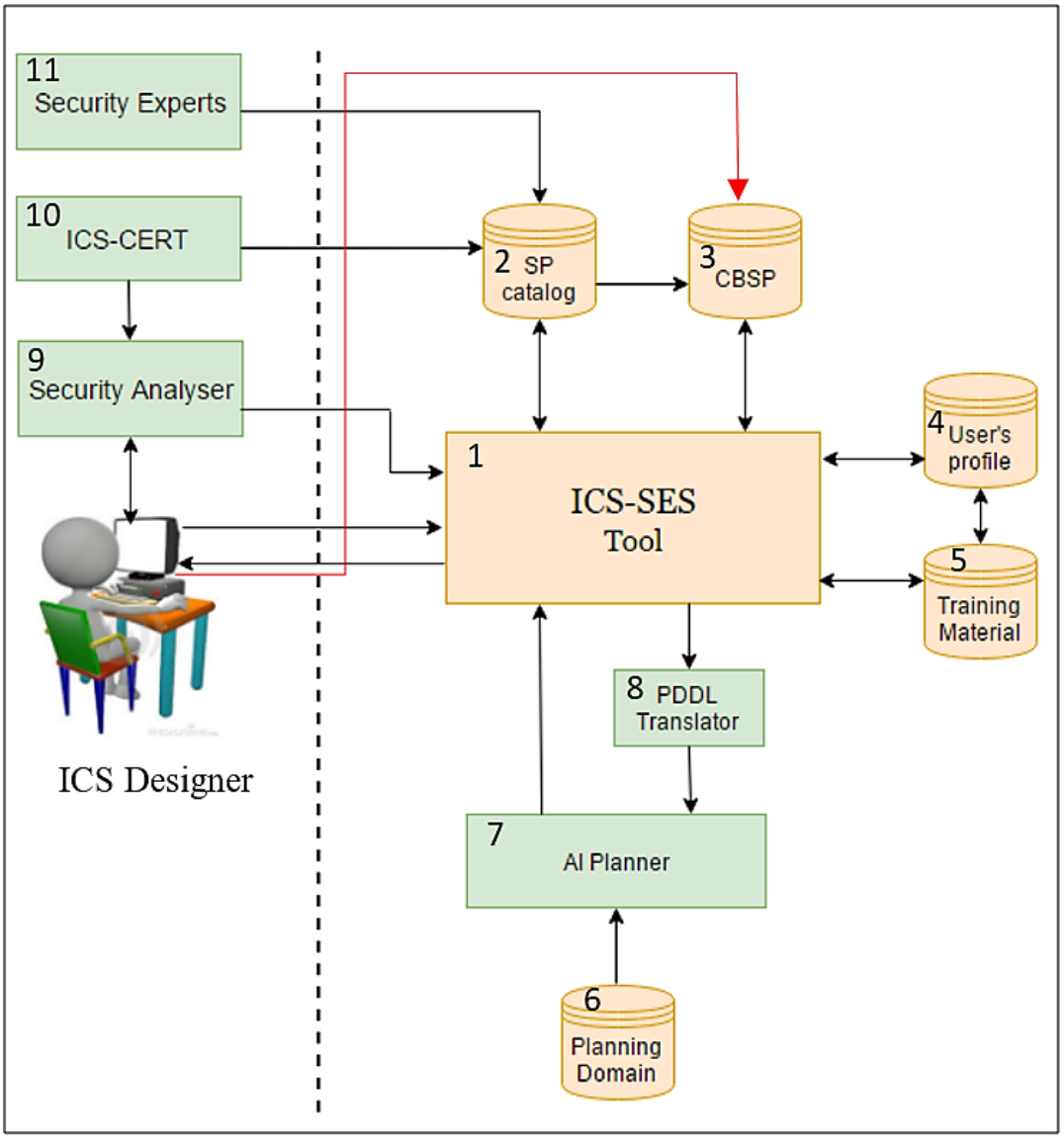


Figure 5-6 ICS-SES Architecture

5.4.1- ICS-SES Tool

The ICS-SES Tool was implemented to integrate the two main parts of the proposed framework, the security guide and training. The tool is intended to be integrated into the development environment by providing a security guidance intervention the moment a security analyser identifies a security flaw in a system model. It implements a user interface to facilitate the interactivity. It receives its inputs from an external security analyser '9' that is integrated into a graphical modelling tool that is used by a system designer to identify any security weaknesses in a system model.

The tool provides a set of suggested security patterns to solve for any discovered security flaw, and allows a system designer to choose a suitable pattern. It also provides learning support by offering adaptive security training material in an interactive manner. In addition, it collects feedback after pattern evaluation through which it updates Case-Based Security Patterns (CBSP).

The ICS-SES Tool communicates with all system components and plays the role of management tool throughout the support process. The tool can be implemented using any graphical programming language.

5.4.2- Security Patterns Catalogue (SP catalogue)

The security patterns catalogue was created by adapting two existed categories: the ICS-CERT category (Nelso and Chaffin, 2011) and the catalogue in reference (Motii et al., 2015), as discussed in Section 5.2.2.1. Figure 5.7 shows the database model diagram of the security pattern catalogue.

ICS Asset: is a component or part of a system that is of value to the ICS and essential for performing its tasks. An asset can be hardware, communication networks, software, or people. In this work, ICS assets were classified into the following categories, as discussed in Section 5.2.2.1:

1. Centre Controller: such as HMI and SCADA server.
2. Field Controller: such as PLC, RTU and IED.
3. Field Units: such as actuators, meters and sensors.
4. Network communication.

ICS asset data table includes the attributes of an ICS asset: asset identifier, asset name and asset category.

ICS Vulnerability: is a flaw in an ICS asset that constitutes a security weakness. All related ICS vulnerabilities were categorised, as discussed in Section 5.2.2.1, and saved in a Vulnerability table. ICSVulnerability contains a vulnerability identifier, vulnerability name, vulnerability category and an ICS asset.

ICS Risk: is a potential attack by a threat that exploits one or more vulnerabilities and has a negative impact that can harm one or more assets. The table of ICSRisks includes a risk identifier, a risk category and the vulnerability that causes the risk. The risk category is linked with the asset table.

Security Requirements: are security properties that reflect the security needs of the ICS system (e.g., confidentiality) in order to mitigate risks. They are used to represent system security objectives. Security requirements were expressed as a table of ICSAsset, related risks and expected security requirements.

Abstract security patterns (ASP): encapsulate security solutions for recurring problems without implementation details, such as access control, authenticator and security logger and auditor. Abstract security patterns were classified based on ICS asset, ICS vulnerability and security requirements. The ASP table consists of the ICS asset, security requirements, and abstract security pattern.

Concrete security patterns: are concrete implementations of security patterns. They are derived from abstract security patterns, and include all their aspects, plus additional aspects related to the specific context to enable designers to apply patterns to a concrete design level. Concrete security patterns are expressed in a table of abstract security pattern, concrete security pattern, pattern name and constraints.

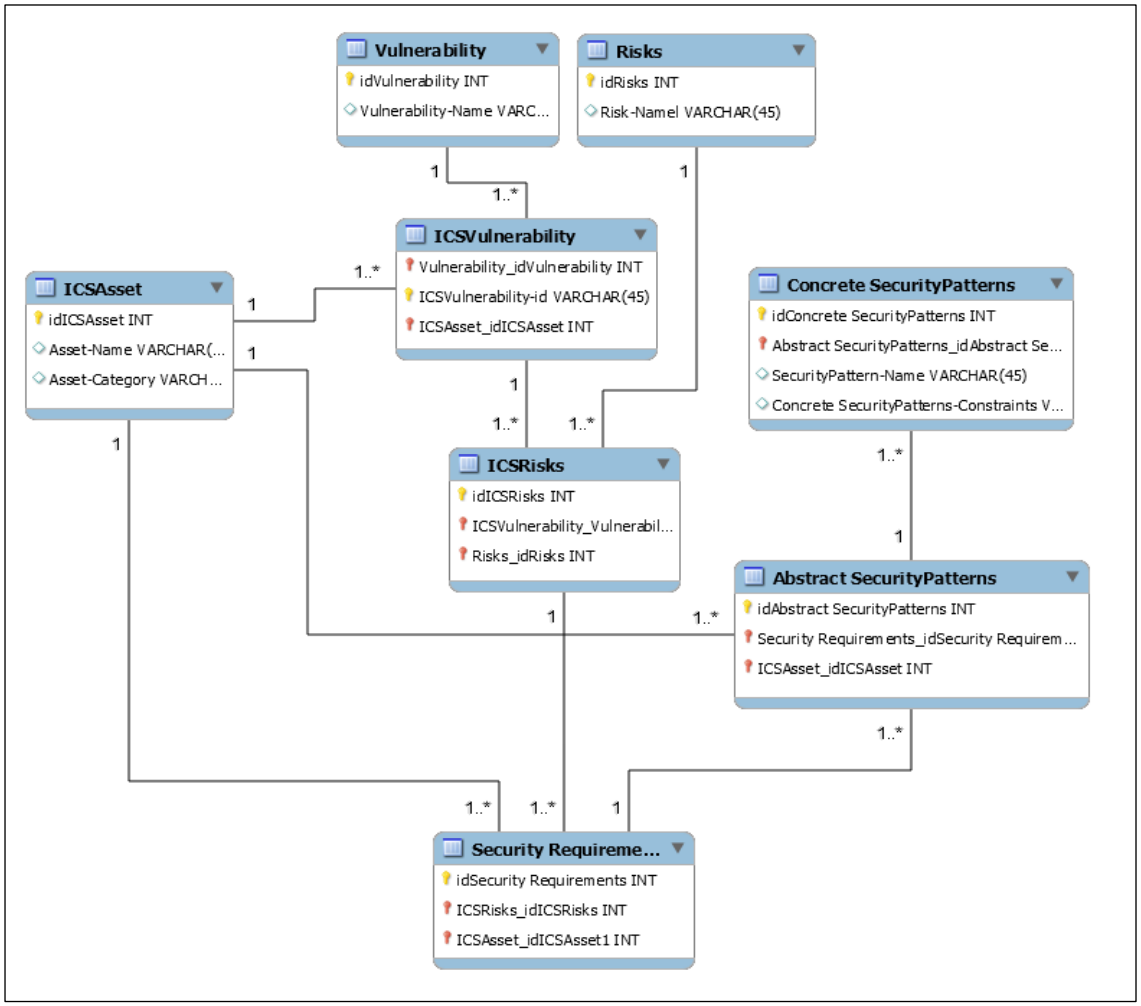


Figure 5-7 Security patterns catalogue data model

For example, Figure 5-8 shows that the vulnerability of unencrypted sensitive data in ICS network communication carries certain security risks:

- Spoofing
- Tampering
- Information disclosure

These risks identify security requirements of:

- Confidentiality of data
- Integrity of data
- Mutual authentication

According to Fernandez *et al.* (Fernandez-Buglioni, 2013), the abstract security pattern that satisfies the above security properties and related to the network asset

was identified as: Virtual Private Network (VPN). This pattern includes two concrete security patterns:

- IP Sec VPN
- TLS VPN

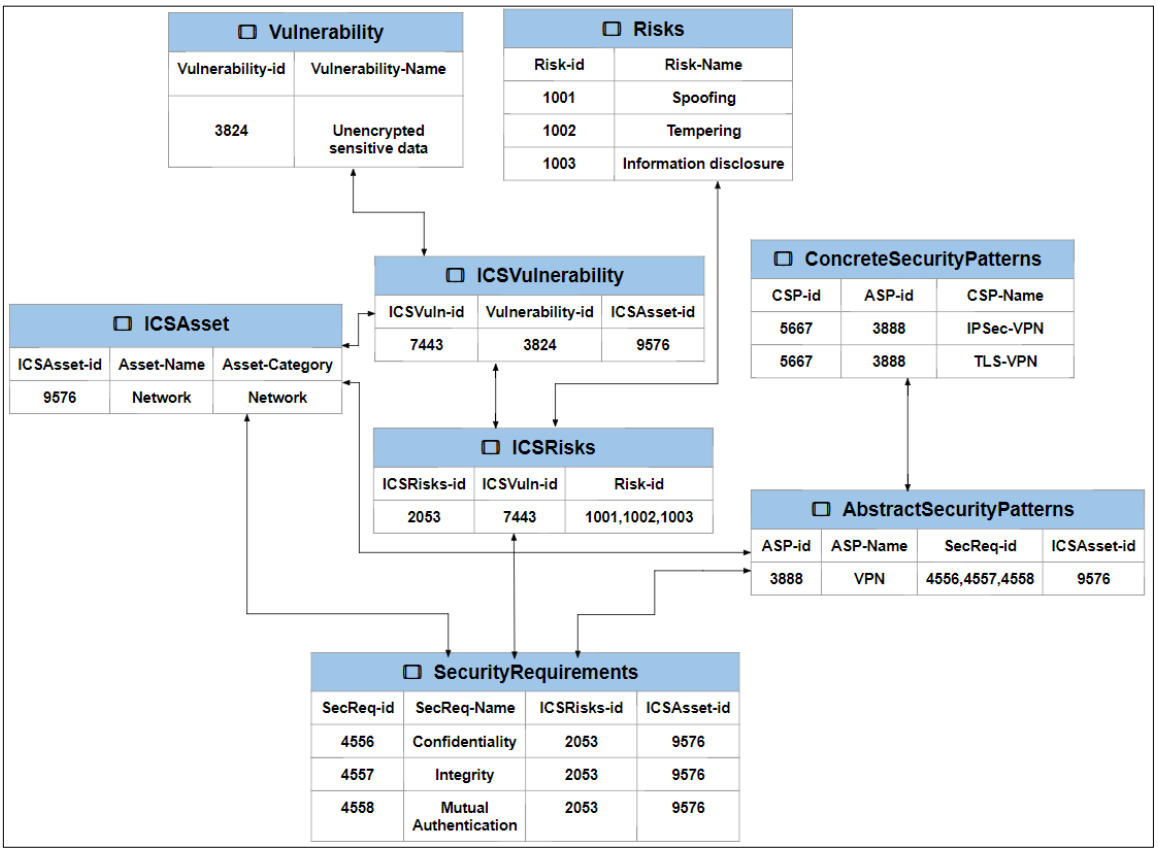


Figure 5-8 Security patterns catalogue data example

5.4.3- Case-Based Security Patterns (CBSP)

Case-Based Security Patterns (CBSP) are derived from the Case-Based Reasoning (CBR) technique that is used to solve problems by retrieving the most similar previous cases from a case base (Aamodt, 1995, Bergmann et al., 2005). CBR is a powerful approach for decision support and solving knowledge-based problems (Aamodt and Plaza, 1994). In our work, a similar technique was used to support the pattern selection process through reasoning by revising past cases. CBSP stores successful pattern selection cases and applications, including all relative information in terms of asset, vulnerability, and security pattern. This information is inserted

through a developer's positive feedback given after evaluating a selected pattern, as explained in Section 5.5.

5.4.4- User's Profile

All developers need to be registered in the ICS-SES system to have individual profiles. A developer's profile includes an e-portfolio, security background and training history. Personal profiles were modelled to facilitate personalised training. Profiles are automatically updated during the training execution process. Once a trainee successfully finishes a learning object, it is added to their personal profile.

5.4.5- Training Material

Security training material was developed based on international standards. Security learning objects were defined and their interrelations were specified, as based on standard meta data. Learning objects were prepared for each security pattern with all related topics. Metadata was defined for each learning object to be translated and used by an AI planner.

Figure 5.9 shows a learning object and its metadata, as defined based on the Learning Object Metadata standard (LOM) specified in the IEEE standards (Hodgins and Duval, 2002).

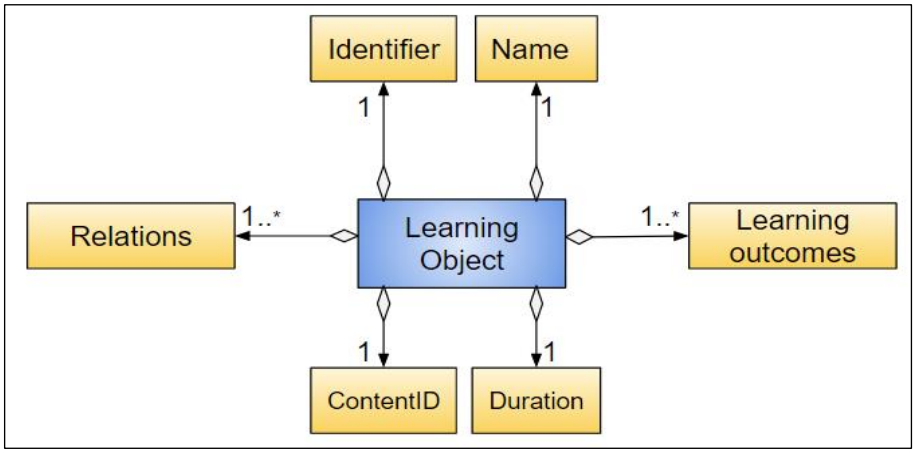


Figure 5-9 Learning Object metadata

Identifier: a unique label that identifies a learning object.

Name: title given to a learning object.

Relations: represents learning object aggregation and dependencies that define prerequisite knowledge levels and competency.

Learning outcomes: the skills and knowledge that are gained upon completion of the learning object.

Content: is the learning activity, e.g., text, diagrams.

Duration: Typical time required to learn the object.

Learning objects were translated into a planning modelling language to be used by an AI planner. The process of metadata translation and knowledge engineering are explained in the next section.

5.4.6- Planning Domain

Our security training method is grounded on the AI planning technique. Thus, all learning objects were mapped into a Planning Domain Definition Language (PDDL) based on reference (Garrido et al., 2009), as explained in Section 5.2.3.2.1, to be used by an automated planner. The training planner generates a training plan tailored to the personal training needs of an engineer by analysing the learning prerequisites and outcomes extracted from the metadata definition.

The planning domain was modelled using a knowledge engineering tool, named itSimple, which is a friendly graphical interface that supports the knowledge engineering process (Vaquero et al., 2007). Figure 5.10 shows a screen shot of the tool. Since 2005, itSimple has been applied in numerous planning applications such as manufacturing (Vaquero et al., 2006), project management (Udo et al., 2008) and petroleum supply ports (Sette et al., 2008). itSimple includes a set of planners (Metric-FF, FF, SGPlan, MIPS-xxl, LPG-TD, LPG, hspsp, and SATPlan) that can be applied to a PDDL model in order to solve a planning problem.

Using the itSimple tool adds an advantage to our method as it can translate an xml metadata file into the PDDL model, which allows for the reuse of available learning material and its incorporation into the learning body of our embedded security

training. Hence, itSimple supports the extensibility of our embedded security training and facilitates the process of updating and adding new objects to our training repository.

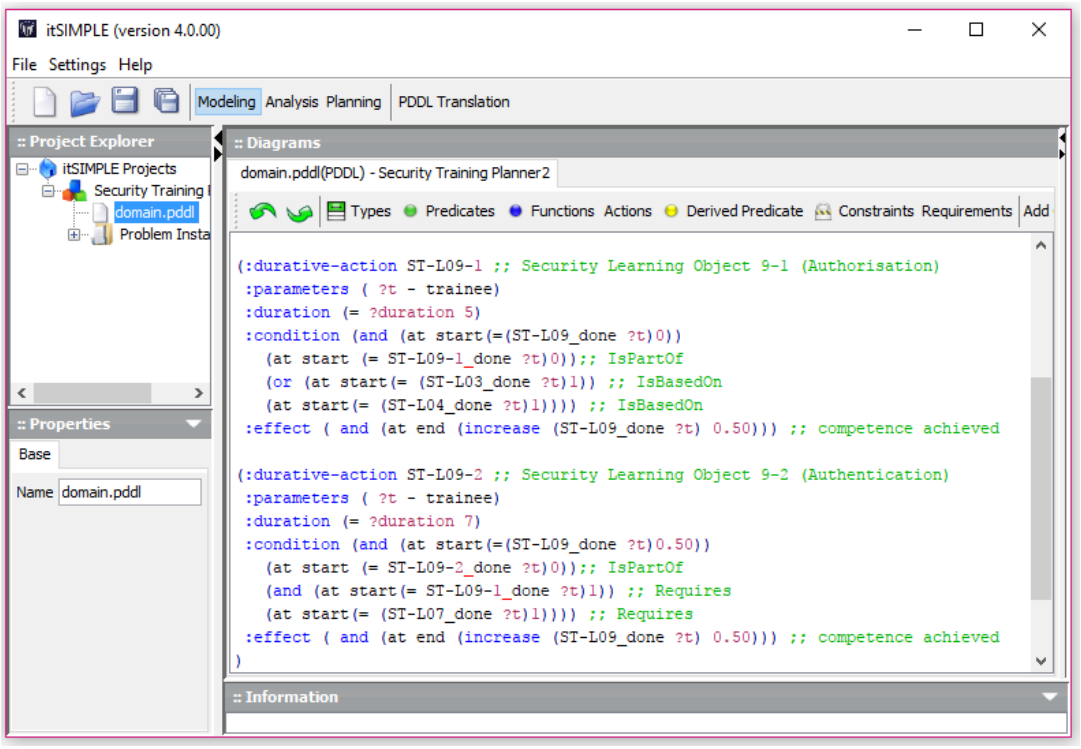


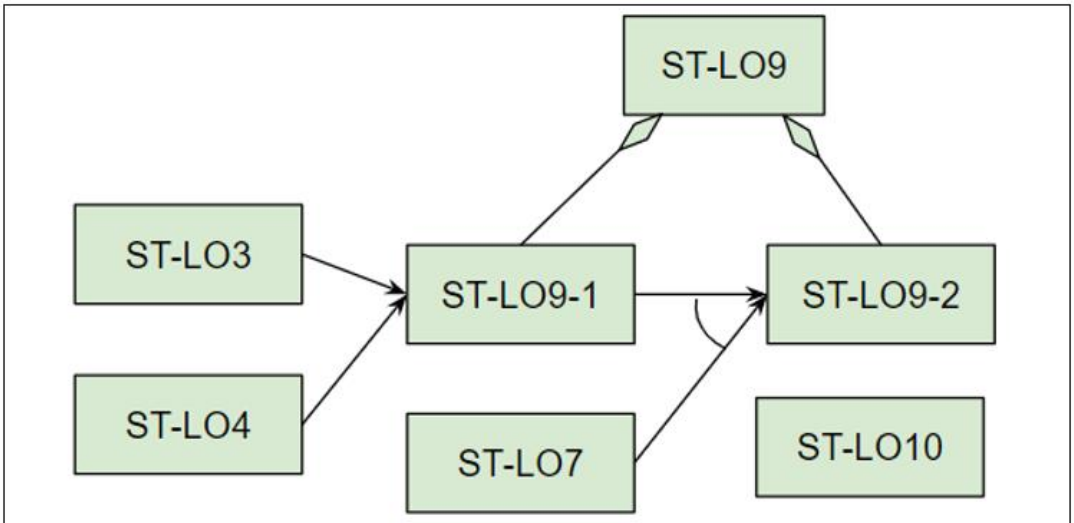
Figure 5-10 Modeling a security training planning domain using the itSimple tool

Learning time was considered an important element of our training method as it was developed to take place during the system design process, 'on-the job training', thus PDDL temporal domain compilation was performed, where time is modelled by means of the artificial fluent total time.

Figure 5.11. depicts an example of modeling learning objects into durative PDDL actions with four entries: parameters, duration, condition and effect. The learning object ST-LO9 consists of two sub-objects, ST-LO9-1 and ST-LO9-2. Only one parameter was defined for a trainee who executes the action. Duration identifies typical learning time. Conditions and effects vary depending on the learning object dependencies and relationships. A numeric function was used in order to deal with different levels of attainment and knowledge. Function 'done' uses a range of values [0-1], where '1' means the learning object has been completed and '0' means it has not been started. It was used to represent different competence levels, for instance, a value of '0.25' means a trainee has learnt only 25% of the learning object. The value is increased when a trainee executes an action and completes the corresponding

training activity. The Start predicate includes the preconditions for the action, and the End predicate represents its effects.

In this example, the ‘ST-LO9-1’ and ‘ST-LO9-2’ learning objects were identified as a part of ‘ST-LO9’, so both objects are necessary to complete ST-LO9. ‘ST-LO9-1’ requires only one action, either ‘ST-LO3’ or ‘ST-LO4’, whereas, ‘ST-LO9-2’ requires the completion of actions ‘ST-LO9-1’ and ‘ST-LO7’. The value of the learning duration is calculated from sub-learning objects. For instance, the duration of ‘ST-LO9’ is ‘12 mins’, which is the summation of its parts’ duration.




```
(:durative-action ST-L09-1 ;; Security Learning Object 9-1 (Authorisation)
 :parameters ( ?t - trainee)
 :duration (= ?duration 5)
 :condition (and (at start(=(ST-L09_done ?t)0))
   (at start (= ST-L09-1_done ?t)0));; IsPartOf
   (or (at start(= (ST-L03_done ?t)1)) ;; IsBasedOn
   (at start(= (ST-L04_done ?t)1)))) ;; IsBasedOn
 :effect ( and (at end (increase (ST-L09_done ?t) 0.50))) ;; competence achieved

(:durative-action ST-L09-2 ;; Security Learning Object 9-2 (Authentication)
 :parameters ( ?t - trainee)
 :duration (= ?duration 7)
 :condition (and (at start(=(ST-L09_done ?t)0.50))
   (at start (= ST-L09-2_done ?t)0));; IsPartOf
   (and (at start(= ST-L09-1_done ?t)1)) ;; Requires
   (at start(= (ST-L07_done ?t)1)))) ;; Requires
 :effect ( and (at end (increase (ST-L09_done ?t) 0.50))) ;; competence achieved
)
```

Figure 5-11 An example of translating learning objects into durative PDDL actions

A planning domain represents a hierarchal structure of learning activities based on their relationships, and helps a planner to find a suitable training path to meet

individual needs. The planning domain was modeled for all learning objects and saved to be used by a training planner to generate a personalised security training plan.

5.4.7- Using an AI planner to generate a training plan

The previous section demonstrated a planning domain model for learning objects. This section discusses the use of an automated planner to generate a training plan by modelling a planning problem and solving it through the compilation of a planning domain and problem.

A planning problem needs to be generated interactively during ICS-SES tool utilisation for each training case. A training case considers a selected security pattern, trainee profile and background. This information is mapped into problem model propositions including those of objects, initial state, and goal state. The object represents a trainee. The initial state represents the trainee's profile and background, including the attained values explained in the previous section. The goal is to attain the learning object of the security pattern selected by a trainee to solve a security weakness detected in a system model.

Figure 5.12 shows a part of the learning objects hierarchy that was modelled into a planning domain. The figure depicts an example of a training case, which includes the trainee's background and needs, using different colours. The blue rectangles illustrate the learned objects whilst the red rectangle presents the target training object that had been identified based on the selected security pattern through the previous phase of our supported tool. Green rectangles represent unlearned topics.

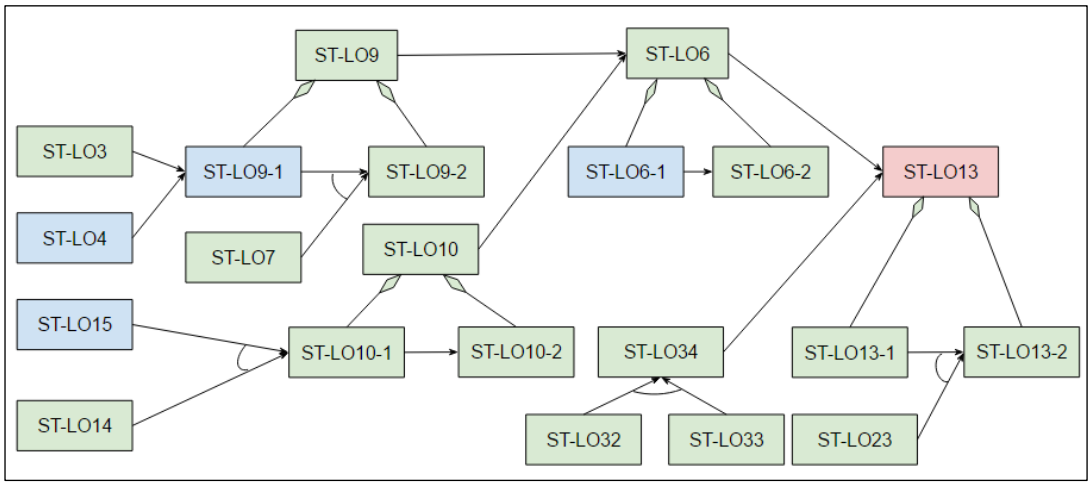


Figure 5-12 A part of the learning objects' hierarchy, relationships, and dependencies

Figure 5.13 shows the corresponding planning problem model of the training case described above. Data was retrieved from a trainee's profile and translated into a planning problem by PDDL translator '8' shown in Figure 5-6. Once an appropriate planning problem was generated, a planner applied it to the predefined planning domain to generate a personalised training plan. The training plan consists of a sequence of learning objects that are executed by a trainee.

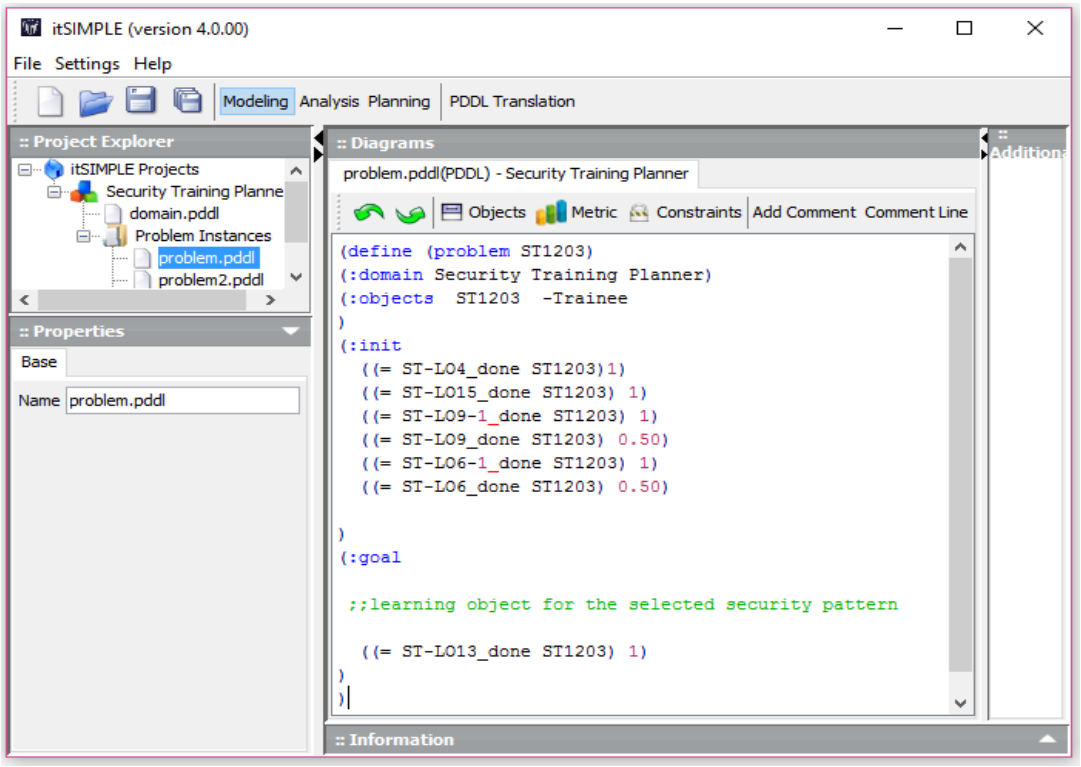


Figure 5-13 An example of a planning problem model

According to the example shown in Figure 5.11, there is more than one possible plan for the given problem as based on the trainee's background. However, the training planner choses the shortest plan, as based on the learning duration identified in the planning domain, to maintain efficiency. For instance, three different plans can be found to solve the above problem: plans A, B and C, as explained in Table 5.4.

Plan A= {ST-LO7, ST-LO9-2, ST-LO6-2, ST-LO23, ST-LO13-1, ST-LO13-2} Plan A consists of six learning objects, required time = 1+1+1+2+1+1= 7 mins. Plan B= {ST-LO14, ST-LO10-1, ST-LO10-2, ST-LO6-2, ST-LO23, ST-LO13-1, ST-LO13-2} Plan B consists of seven learning objects, required time = 2+1+1+1+2+1+1= 9 mins. Plan C= {ST-LO32, ST-LO33, ST-LO23, ST-LO13-1, ST-LO13-2} Plan C consists of five learning objects, required time = 3+3+2+1+1= 10 mins.

Table 5-4 An example of training plans

Figure 5.14. shows that the training planner generated the shortest plan, ‘A’. The training plan is sent to the ICS-SES tool, which then retrieves the corresponding training activities from the training material repository and delivers them to the trainee.

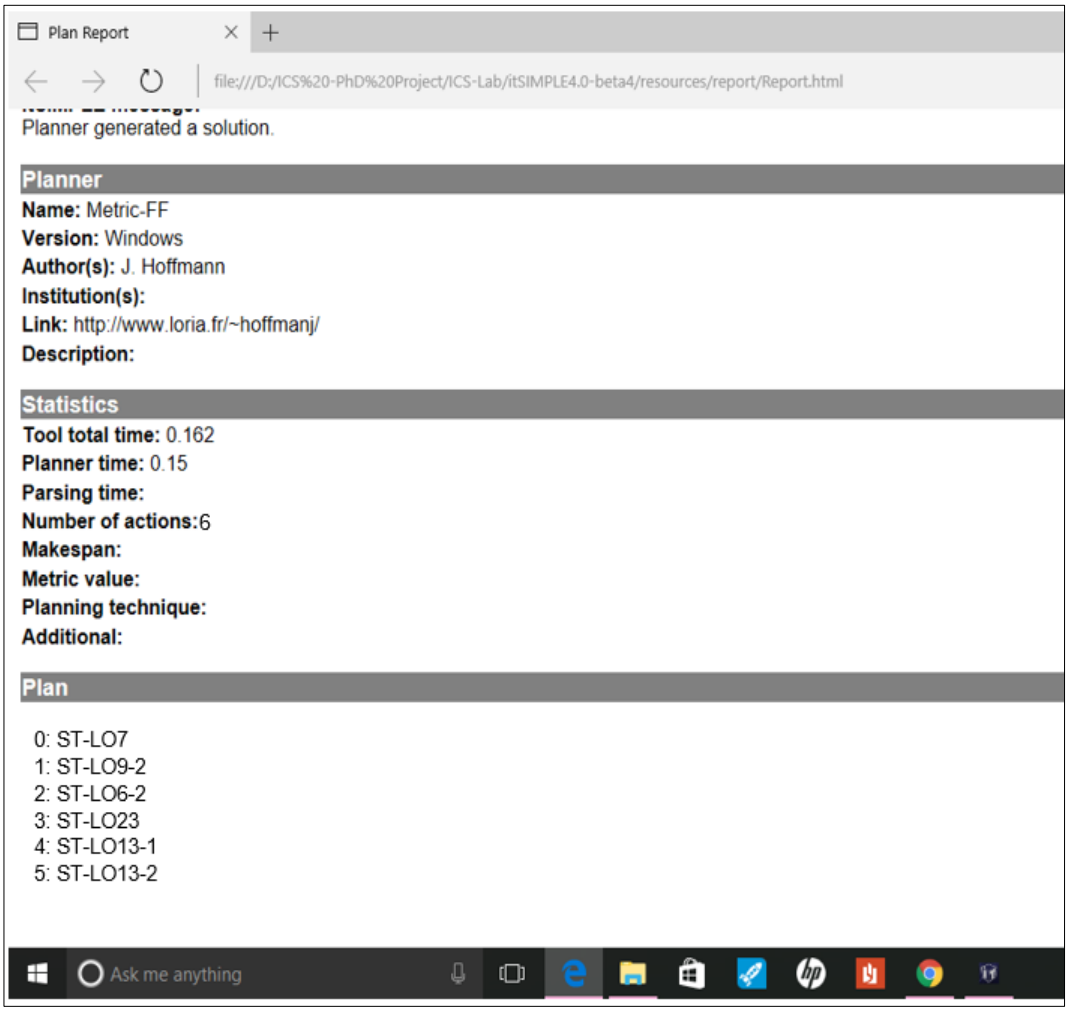


Figure 5-14 A training plan generated by a planner

ICS-SES architecture, shown in Figure 5.6, collaborates with external resources including those of ICS-CERT and security experts. The ICS-CERT database extends our security pattern catalogue by identifying new vulnerabilities, as was discussed in Section 5.2.2.1. Security experts extend our repository of security patterns by developing new security solutions in the form of security patterns.

5.5- ICS-SES workflow

Our security support flow is demonstrated in Figure 5.15. As mentioned previously, ICS developers need to be registered with the ICS-SES system in order to have individual profiles. The figure shows that a logged developer starts using the ICS-SES system after detecting a security problem after scanning a system model during the system design phase. If a developer decides to use the ICS-SES tool, the security guide helps to solve the problem by performing the four pattern selection steps discussed in Section 5.2.2.2, and providing a set of related security patterns. When a

developer chooses one of the patterns, the security training tool generates an adaptive training plan through the three learning phases discussed in Section 5.4 that is tailored to the selected pattern and the developer's needs. After each training object execution, the training planner generates a new plan, based on the last update of a developer's background, in order to improve the efficiency of the training plan. When a training goal is obtained, which means learning the selected security pattern, a developer configures the selected pattern to a system model using a modelling tool and evaluates it using a security scanner. Then, the developer is asked to give feedback as to whether the pattern solves the problem that was discovered. Positive feedback is stored in CBSP, as introduced in Section 5.4.3, and used as a recommendation in the next pattern selection process that shows the same problem context. The entire support flaw keeps recurring until the problem is solved or a developer chooses to exit.

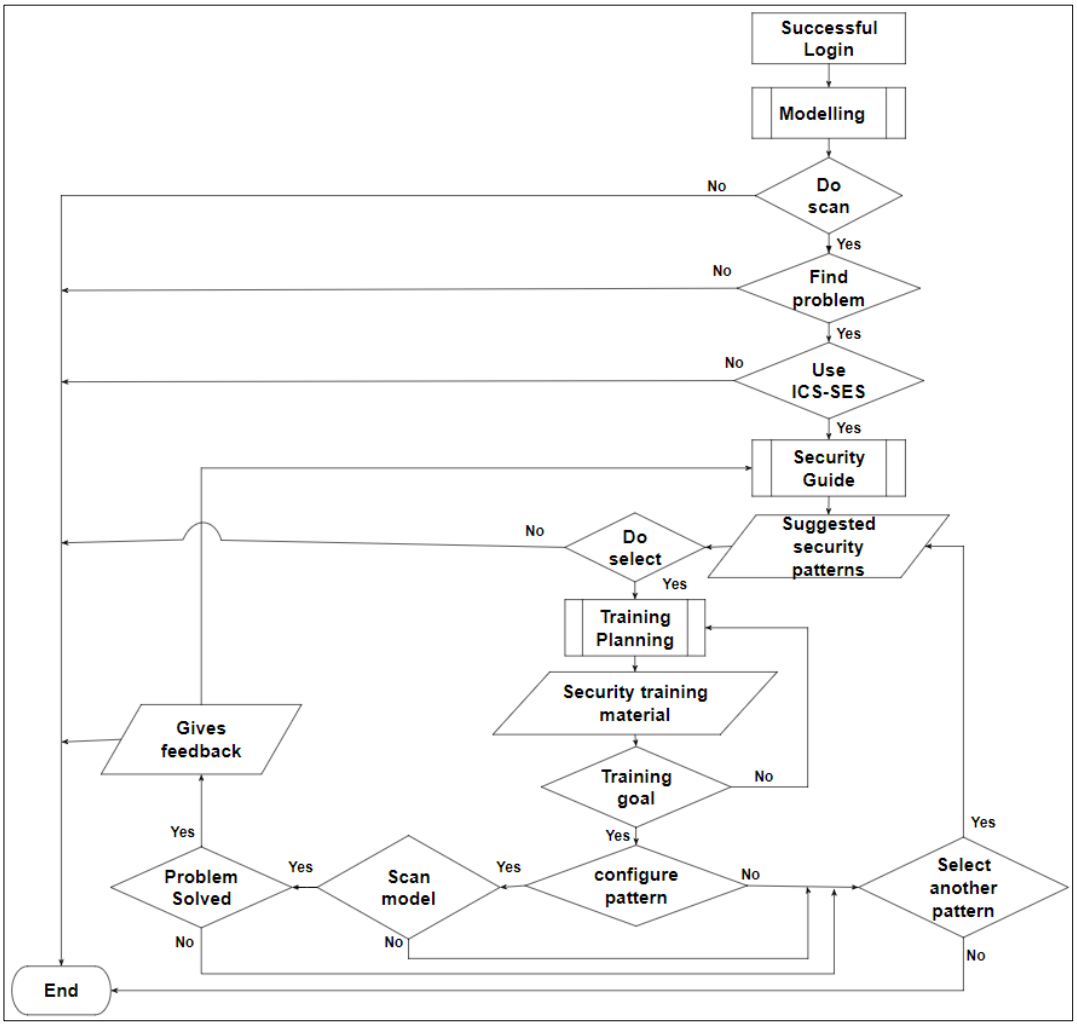


Figure 5-15 ICS-SES Flowchart

5.6- Conclusion

This chapter has presented a novel framework, named ICS-SES, that can be used to support developers in the design of secure control systems. The ICS-SES framework focuses on two support dimensions, technical and learning, to assist engineers in developing ICS security by design within their work environment.

The framework combines two new support methods, namely a pattern-based security guide, which guides pattern selection to solve a design security problem, and embedded security training, which provides security training material tailored to the problem context and the personal training needs of a system designer.

The chapter initially presented the approaches and techniques that motivated the design of our particular framework. Then, the two main methods underpinning the ICS-SES framework were explained and system requirements were outlined, followed by demonstrating the ICS-SES architecture and discussing its components. Lastly, the entire ICS-SES workflow was explained with a demonstrative flowchart.

The feasibility of the ICS-SES framework needs to be tested by showing that it can support engineers in the design of secure systems and in improving their security knowledge. The next chapter introduces the ICS-SES prototyping tool and discusses the design of a controlled experiment to evaluate the effectiveness of the ICS-SES tool.

Chapter 6
A Controlled Experiment for Evaluating ICS-SES

Chapter objectives

- To introduce the controlled experiment method to evaluate ICS-SES.
- To outline the purpose of the experiment
- To demonstrate the experiment design and procedure
- To discuss the preliminary results of the experiment
- To illustrate the experimental execution

6.1- Introduction

Following Chapter 5, which introduced a supported method, namely ICS-SES, to assist developers in designing secure control systems, this chapter uses a prototyping tool for ICS-SES to empirically evaluate its usability and effectiveness in supporting ICS security by design. Before carrying out the evaluation of ICS-SES, it was necessary to clearly identify the evaluation methodology. A controlled experiment, which is one of the evaluation methods used in design science research, was chosen to evaluate the qualities of our artefact 'ICS-SES'. The selection of this method was discussed and justified in Chapter 3, Section 3.4.4.

Using a controlled experiment methodology allowed the researcher to conduct a focus study that produced statistically significant results. It helped her to emphasise specific variables and measure the relationships among them. It was useful in formulating the study hypotheses through the clear definition of the questions being studied throughout the experiment. Such an evaluation method usually results in well-defined dependent and independent variables and well-defined hypotheses (Basili, 2007). A controlled experiment procedure was designed and underwent preliminarily testing before executing the main evaluation study to ensure the viability of the procedure.

The chapter is divided into the following sections. Section 6.2 highlights the purpose of the experiment. Section 6.3 discusses the experimental design, demonstrating the variables, materials, tasks and participants. Section 6.4 explains the experimental procedure. Section 6.5 highlights the preliminary results and discusses the

consequent changes. Section 6.6 demonstrates the study execution. Section 6.7 gives a summary of the chapter.

6.2- The Purpose of the Experiment

An experiment was designed to evaluate whether the ICS-SES tool can help engineers to develop ICS security by design and improve their security knowledge. The main objective of the experiment was to ascertain the usefulness and effectiveness of the security guidance and learning provided by the ICS-SES tool, and to check whether using the tool can help engineers to design secure control systems. The focus of the experiment was on the use of the ICS-SES tool to provide pattern-based solutions and tailored learning materials to solve a particular security problem in a system model. Therefore, the experimental evaluation of the ICS-SES tool was carried out to test the following hypotheses:

H1- Effectiveness (Performance): Participants will be better able to solve any security problem(s) in a system model with the help of the ICS-SES tool.

H2- Effectiveness (understanding the problem): Participants will better understand the security problem(s) with the support of the ICS-SES tool.

H3- Effectiveness (understanding the solution): Participants will better understand the security solution(s) with the help of the ICS-SES tool.

H4- Ease of task (ease of solving the problem): The difficulty in solving the security problem will be reduced with the help of the ICS-SES tool.

H5- Efficiency (Time): The time taken to solve the security problem, given in the scenario, will be reduced when using the ICS-SES tool.

The above will be in addition to the qualitative data obtained from participants' feedback.

6.3- Experiment Design

The experiment was designed based on the practical methodological guidance provided by Ko*et.al*. in the evaluation of software engineering tools with human participants (Ko et al., 2015). The controlled experiment was designed to answer the

research question ***"Can a supported tool assist developers in designing secure control systems?"***.

There was only one treatment of the 'ICS-SES tool' used in the experimental design. The comparison study was between an experimental group, named the supported group, who used the ICS-SES tool and another, named the plain group, who used a conventional development environment that was replicated by a graphical tool, the 'Plain tool'. The key property of the experiment was that both groups received the exact same materials including tutorials, problem scenario, tasks, instructions, IDE and experimental environment; for the only difference was that our supported tool was only provided to one experimental group to identify whether there was any difference in the outcomes of the two groups.

6.3.1- Ethical Approval

As the experiment involves human participants, it was essential to secure an ethical approval application before conducting this study to ensure that it adhered to British Psychological Society (BPS) ethical guidelines. The ethical approval for the experiment was granted by the Faculty Research Ethics Committee (FREC) (ref:1415/247-1) (Appendix B-2). It covered issues related to respect of participants; confidentiality of the collected data and identity of participants; standard of self-determination, so participants can withdraw partially or completely from the experiment at any time and without explanation; and honesty and accuracy when representing the collected data.

6.3.2- Experiment Variables

6.3.2.1- Independent Variable

The ***ICS Security Engineering Support (ICS-SES) tool***, which is a prototyping tool, was developed and used in the experiment, and is introduced in Section 6.3.3.1.

6.3.2.2- Dependent Variables

The experiment focused on determining whether participants could use the ICS-SES tool to secure a system model by mitigating a security vulnerability and learning new

security skills. In particular, the tool's usability was evaluated in terms of the effectiveness, efficiency, learning outcomes and ease of the task compared to the conventional development environment that was replicated by using the 'Plain tool', as presented in Section 6.3.3.2.

Performance, the performance of the participants in solving the problem is reported and measured to assess the effectiveness of the tool in supporting the participants in designing secure systems.

Learning outcomes, the understanding of the problem and the solution were measured pre- and post-experimental task to evaluate the effectiveness of our tool in improving participants' knowledge.

Time, the time spent to complete the task of solving the problem is reported and measured to evaluate the efficiency of our tool. A time evaluation was considered due to the nature of our tool, as it is intended to be used within the workplace, where time is a significant issue.

Participants' feedback, which was obtained from the post-questionnaires, is measured to evaluate the ease of performing the experiment task.

6.3.2.3- Controlled Variables

Participants, the participants were undergraduate and postgraduate students from De Montfort University, Leicester, UK. The selection and size of the sample is discussed and justified in Section 6.3.6.

Tasks, the experiment session lasted for a maximum of one hour. Participants were required to solve a security problem in a system model using either the ICS-SES tool or Plain tool and complete a pre-questionnaire and post-questionnaire.

6.3.2.4- Extraneous Variables

Experience with security patterns and ICS security were defined as extraneous variables in this experiment. Engineers who have security experience will have an obvious ability to understand and solve security problems without the help of the

ICS-SES tool. Hence, engineers with security experience were excluded from the experiment.

6.3.3- Experiment Material

6.3.3.1- ICS-SES Tool

Having developed the proposed supported framework, which was introduced in the last chapter, it was necessary to further develop a prototyping tool to evaluate this framework. The prototype, which was initially proposed in the early 1970s, typically simulates certain aspects of the final system or product (Grimm, 2004). In this work, the ICS-SES tool was developed to evaluate the feasibility of our supported framework.

The ICS-SES tool was implemented as a Java application based on the ICS-SES architecture, as presented in Chapter 5, Section 5.4. ICS-SES components were implemented using Eclipse, MySQL for database management, and the itSimple tool for knowledge engineering and planning.

The tool was built to provide a graphical user interface with emphasis on two main functionalities: first, the is tool intended to guide users in selecting security patterns that solve a security problem; the second was to provide personalised training material related to the security problem and tailored to the user's background.

Pattern-based security guide - the selection method, which was demonstrated in Chapter 5, Section 5.2.2, was implemented to extract a set of secure design pattern candidates from the security patterns catalogue based on the security problem in a system model. For example, Figure 6.1 shows the ICS-SES Tool providing a set of suggestions to mitigate an unauthorised access vulnerability. The tool allows users to choose one of the pattern candidates and depicts the corresponding changes in the system model before confirming their modifications. Based on our selection method, authorisation was identified as an abstract security pattern candidate that includes four concrete patterns: Access Matrix, Multilevel Security, Role-Based Access Control (RBAC), and Attribute-Based Access Control (ABAC). Concrete patterns are displayed in a combo menu to allow users to select a suitable pattern according

to their preference. Figure 6.2 shows that how RBAC pattern would be configured to ‘Siter1PLC’ in the system model and the tool allows users to confirm these changes.

Figure 6-1 ICS-SES tool provides a set of pattern candidates to solve the security problem

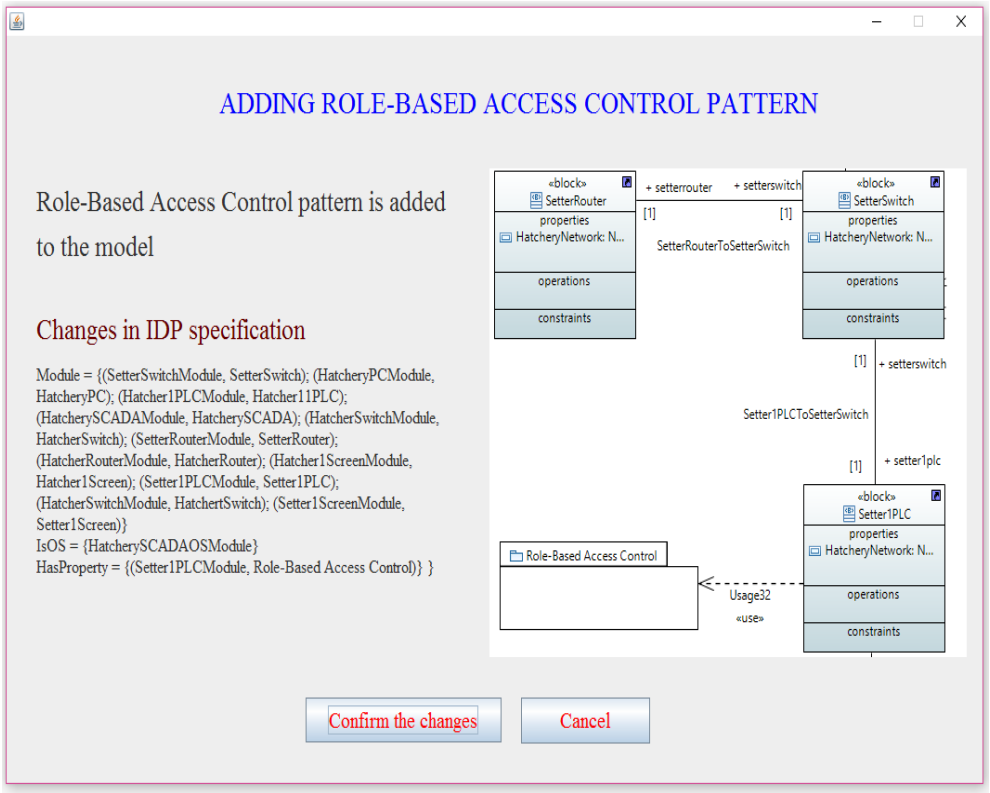


Figure 6-2 ICS-SES tool displays the corresponding changes in the system model after selecting Role-Based Access Control (RBAC) pattern

As discussed regarding ICS-SES architecture, Chapter 5, Section 5.4, our method used an external security analyser to identify security weaknesses in a system model, which was then used as an input to the ICS-SES tool. The analyser developed by (Lemaire et al., 2015) was chosen for several reasons: first, the analyzer was developed as a plugin integrated into the Papyrus modelling tool to be used at the design stage. Second, it is based on analysing a control system that was modelled in SysML, which is commonly used to develop ICS, by parsing the model into Imperative Declarative Programming (IDP) and using logic theory to extract vulnerabilities. Third, the knowledge-base that was used in the analyser is inherited from the ICS-CERT database used by our tool. However, the analyser is still under development, and cannot automatically translate a system model into the IDP file to be analysed. Any changes in the system model need to be written immediately into the IDP file to be considered in the analysis process. Therefore, it was not possible to use the analyser in our experiment, as system engineers are not familiar with IDP, and providing IDP training is outside our scope, and more practically would make the experimental time significantly longer.

On the other hand, there was a need to evaluate the security pattern that was suggested by our tool and chosen by a participant. In order to overcome this challenge and prepare an experimental environment that replicates how our tool would work in a real-world environment. The scenario of a system model that had a security flaw was prepared, and showed the report produced by the analyser to participants through our tool. All cases of the patterns' configuration were prepared and saved in the ICS-SES tool, which displays them according to the user selection. The solution was also defined to assess the pattern selection. When a participant chooses a pattern from a set of pattern candidates, the ICS-SES tool discovers whether that pattern actually solves the problem by comparing it with the pre-defined solution and then providing a suitable message, as shown in Figure 6.3 and Figure 6.4.

Figure 6.3 shows that the pattern chosen by an engineer did not solve the problem, as the analyser is still giving the same result for the problem. This means the engineer needs to try another pattern.

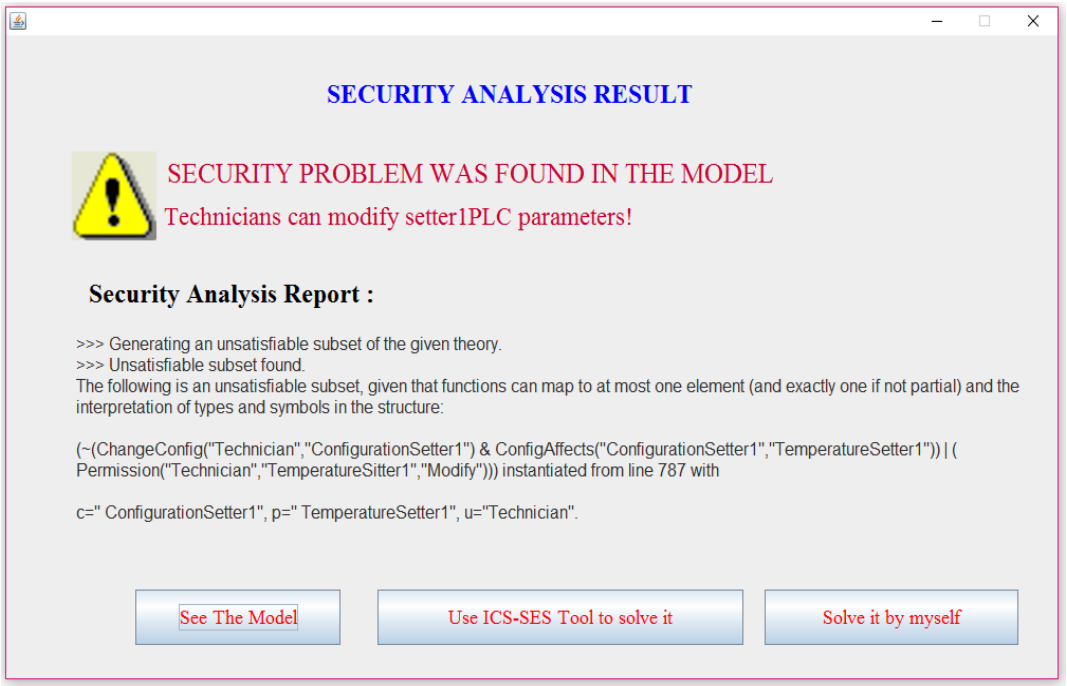


Figure 6-3 Result from the security analyser shows a security problem

Figure 6.4 shows that the pattern chosen by an engineer solved the security problem as the analyser no longer detects the problem after pattern configuration. The message asks to notify the experiment conductor for efficiency evaluation, as discussed in the experimental procedure in Section 6.4.

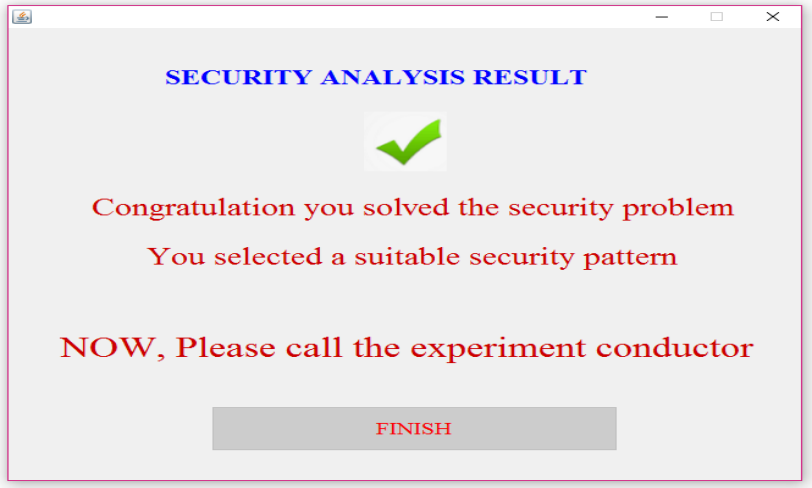


Figure 6-4 The message notifying the selection of a suitable security pattern

Tailored security training- is the second main functionality provided by the ICS-SES tool that implements the training method, as demonstrated in Chapter 5, Section 5.2.3, to provide security training material tailored to the personal needs of the tool user. The training is planned based on the design context, which is related to the

selected pattern and the prior knowledge of the user. The ICS-SES tool offers the security training material that had been specially prepared by the researcher based on online training resources such as ICS-CERT and NIST. Each learning object was presented on a page in the Google site to facilitate training tracking using Google Analytics, as discussed in the next chapter. Figure 6.5 shows some examples of training material related to the Role-Based Access Control (RBAC) pattern.

In our training method, the training needs assessment is typically based on data retrieved from a trainee's profile. However, using the question/answer method can uncover any knowledge that has been gained informally such as through self-learning and informal discussion. It was sufficient for our prototype to use question/answers to check a trainee's understanding of a topic because of the need for a reasonable number of participants and as the nature of the study meant that suitably diverse training needs could not be adequately generated.

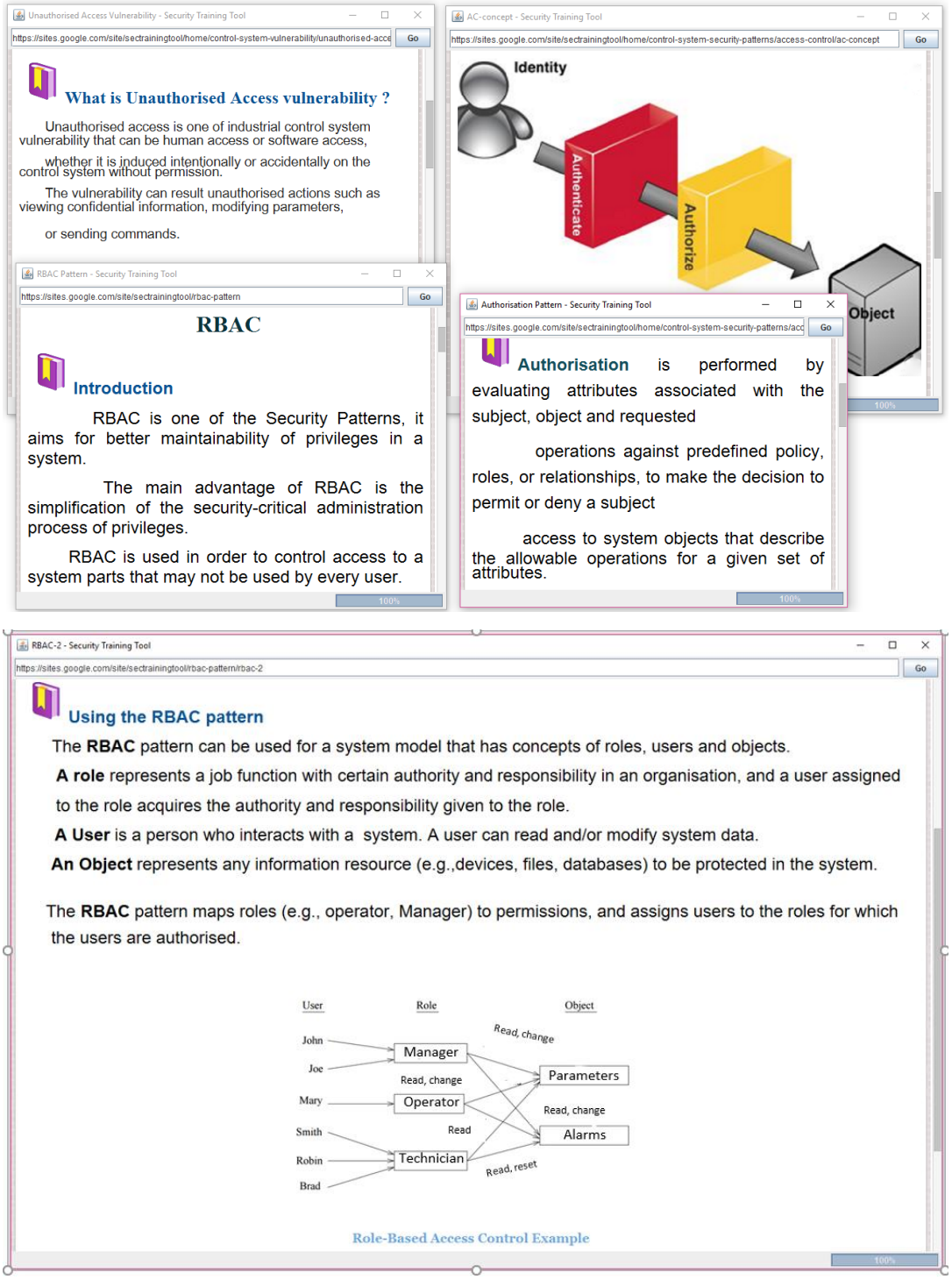
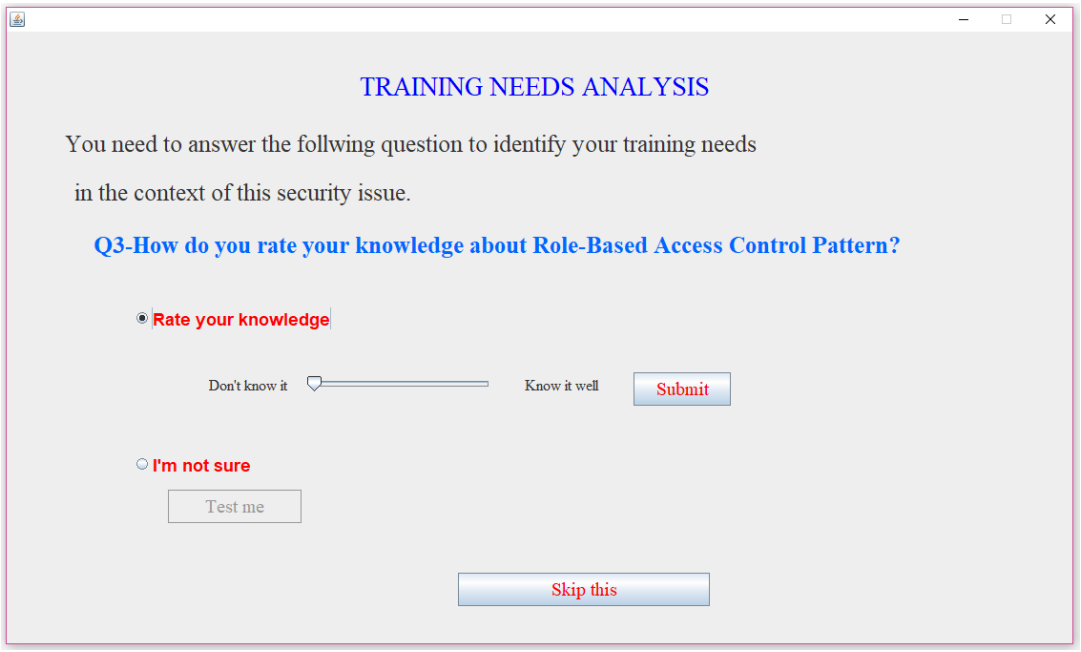


Figure 6-5 Training material for Role-Based Access Control (RBAC) pattern

When a participant chooses a security pattern from the list of patterns suggested by our tool, questions are asked regarding every related learning object in order to assess the user's prior knowledge and accordingly provide personalized training, as shown in Figure 6.6. Participants have the option to rate their knowledge, or let the tool test their knowledge, by asking some related questions to determine the user's level of understanding. Based on the level of knowledge, this resulted in either self-rating or

tool assessment, with the tool providing tailored training material. This process is iterative for all training topics related to the selected security pattern.



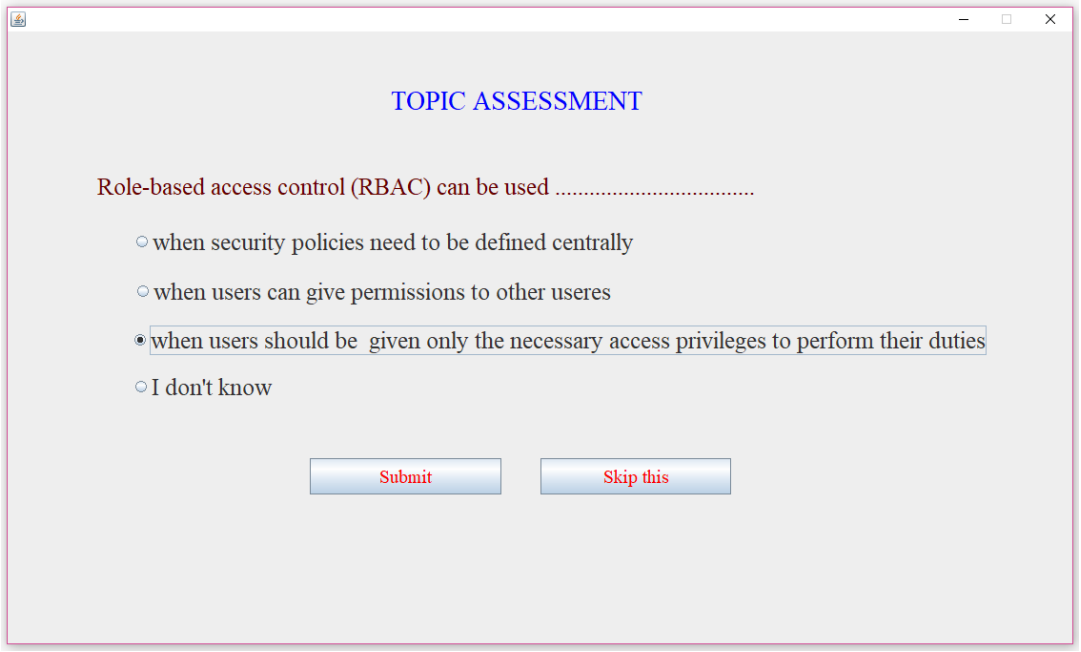


Figure 6-6 The tool test for training needs assessment

In the case of self-rating, the knowledge rate is divided into four main levels: novice (0-25%), intermediate (26-50%), advanced (51-75%) and expert (76-100%). Accordingly, the tool determines the amount of information that is required for any given user. The tool offers the whole topic to novice users, an overview for intermediate users, brief information that presents only the main points of the topic

for advanced users, and no information for experts. However, all users have the option of extending the training material to more detailed information.

If users choose to be tested by the tool, they would be asked a number of questions about each sub-topic. Users' answers were used as the basis for the training planning loop, where the information for the corresponding sub-topic is shown when users get any given question wrong.

6.3.3.2- Plain Tool

A graphical tool, named the 'Plain tool', was developed so the control group, the 'Plain group' would have the same working environment as the experimental group. The Plain tool allows participants to choose a solution pattern for any security weakness identified in the system model. It also allows them to see the corresponding changes after pattern configuration and check whether the vulnerability has been mitigated. The tool provides exactly the same development environment as the ICS-SES tool except for the associated security guidance and training. Figure 6.7 shows a screenshot of the Plain tool. The tool allows the user to choose a solution from a security pattern catalogue and evaluate it, though without the support of our ICS-SES tool. The tool was developed to simulate the real development environment where engineers design control systems without any support of security engineering.

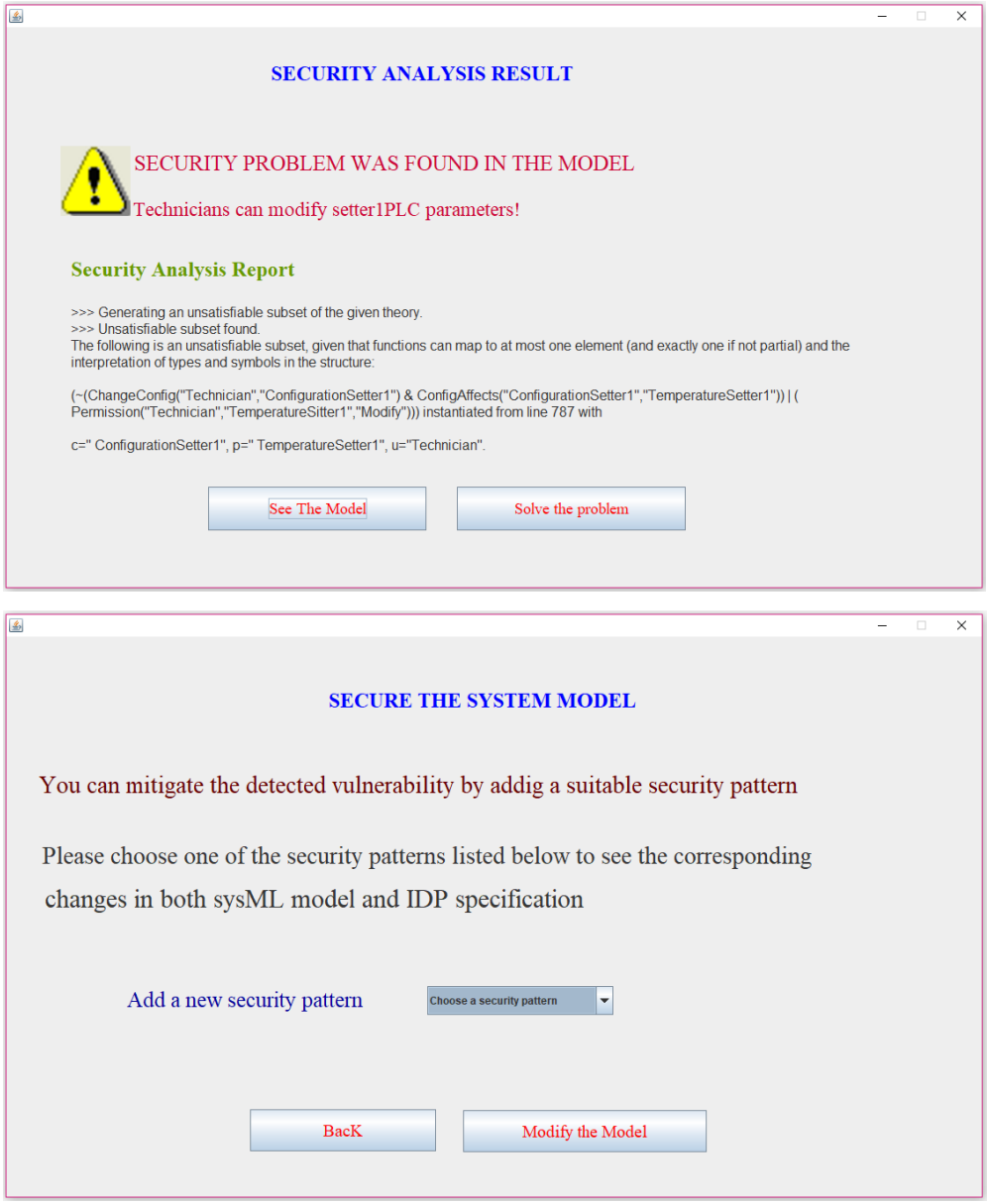


Figure 6-7 The Plain tool, as developed for the control group

6.3.3.3- Pre-Questionnaire

A pre-questionnaire was designed in order to understand how engineers currently design control systems in terms of security engineering, and to know about their security training history. It started with an identifying participant reference, which was a unique code provided by the researcher, that linked individual responses for the pre- and post-experiment questionnaires for comparison purposes. The questionnaire was divided into four sections, as presented in Appendix D-1.

Section-A- Security Background

This section aimed to identify participants' backgrounds and their awareness of ICS vulnerabilities, security patterns and guidelines, and their role in developing secure systems. This section helped to understand the relation between what participants think about their level of security knowledge and their actual performance in solving a given security problem.

Section-B- Security Training

The section aimed to identify participants' security training histories to determine whether participants had security experience, as this was used to filter participants. Participants who had good security knowledge were excluded from the experiment.

Section-C- User Motivation

This section was designed to discover engineers' motivations regarding the construction of secure systems and improving their security skills. The section also helped to understand what may increase and decrease their satisfaction when using our tool.

Section-D- Pre-test

This section was designed to objectively test participants in terms of information security. The purpose of the pre-test was to ascertain whether participants had any pre-existing knowledge related to the security problem given in the scenario. The pre-test and post-test method is widely used for comparing groups or measuring the effectiveness of experimental treatment (Dimitrov and Rumrill Jr, 2003). The section involved two multi-choice questions to allow the assessment of participants' backgrounds regarding the two main aspects of the security problem and its solution. The result of the pre-test was compared with that of the post-test, which included the post-experiment questionnaire, to assess the outcomes of using the ICS-SES tool.

6.3.3.4- Post-Questionnaire

Two different versions of the post-questionnaire were designed for the two groups of participants, the experiment and control groups, to collect information about

participants' experiences in terms of solving the given security problem. Data collected from two groups was compared to test the hypotheses defined in Section 6.2. Participants were also asked to use the same references that were used in the pre-questionnaire to determine individual performances and changes.

Post-questionnaire (A) for Supported group

This questionnaire was developed for the group using the ICS-SES tool to collect information about their experiences and identify the usability of the tool. It was divided into four sections, as shown in Appendix D-2.

Section-A- Participants' Experiences

This section was aimed at collecting information about participants' performances in the task of solving the security problem given in the system model, and identifying their experiences in solving it, and in understanding the problem and its solution.

Section-B- Usability Evaluation Framework Cognitive Dimensions (CD) of the ICS-SES Tool

This section was designed as based on the Cognitive Dimensions (CD) framework. CD is an analytical theoretical framework for usability evaluation (Green and Petre, 1996). The validity and reliability of this technique has been assessed by a number of researchers (Kutar et al., 2002) (Triffitt and Khazaei, 2002) (Blackwell and Green, 2000). For our study, six dimensions were used to evaluate the usability of the ICS-SES tool including visibility, difficult mental operations, diffuseness, closeness of mapping, consistency and role expressiveness.

Section-C- Usefulness and Satisfaction

This section was created based on a Usefulness, Satisfaction and Ease of use (USE) questionnaire (Lund, 2001). Participants were asked about the usefulness of our tool in support of security engineering. The questions were asked to determine whether participants were satisfied with the levels of support offered by the tool. In addition, open questions were asked to collect suggestions and recommendations for improving the tool.

Section-D- Post-test

The post-test was created with the same questions as the pre-test. The purpose of the post-test was to assess the knowledge improvement in the context of the given security problem. The answers from both tests were compared to determine the effect of using our tool.

Post-questionnaire (B) for Plain group

A different post-questionnaire was developed for the control group, who did not use our tool. The questionnaire was designed to collect information about participants' experiences on securing a system model by solving the given security problem. The questions were structured into four sections, as shown in Appendix D-3.

Section A: Participants' experiences

This section was designed to collect information about participants' experiences regarding the task of solving the security problem. It was aimed at determining whether they could solve the problem without the support of our tool. The section was also aimed at determining if there were any supportive resources that helped participants to perform the task of securing the system model. Data collected in this section was compared with section-A in post-questionnaire (A).

Section B: Difficulties

In this section, participants were asked a number of questions about the task load to understand the difficulties in solving the problem without the support of our tool.

Section C: Support Needs

This section was created to identify the further support required for developing ICS security by design in line with the previous finding of the needs assessment study presented in Chapter 4.

Section D: Post-test

Participants from both groups were asked the same questions in order to evaluate the outcomes of using our tool through a direct comparison of the results from the two groups.

6.3.3.5- Tutorial

A tutorial was prepared to demonstrate the scenario of a control system that included a security weakness, as presented in Appendix D-4. Participants from both the experiment group and control group attended the same tutorial session to ensure that they received the same information background regarding the scenario. In addition, the last section of the tutorial introduced and demonstrated the ICS-SES tool to the Supported group.

6.3.4- The Problem Scenario

A scenario of a system model, which has a security flaw, was given to participants to compare how they solved the problem with and without the use of our tool. The scenario description is given in Appendix D. It includes a control system description with its SYSML model and the result of security analyser, which highlights a security problem in the model. The scenario was explained and clarified throughout the experimental session.

6.3.5- Experiment Task

This research focusses on supporting engineers in designing secure control systems through guiding them as to the selection of a suitable security solution and improving their security knowledge. Therefore, the main task of the evaluation experiment was to identify a suitable security pattern that mitigated the security vulnerability given in the scenario.

In the Supported group, participants were asked to solve the problem with the help of the ICS-SES tool. They had to choose a solution and evaluate it using our tool. The Plain group were provided a list of all security patterns through the Plain tool. Participants from the latter group had to solve the problem without the help of our tool. However, they could use any other materials they wanted to help them perform the task. A 30 minute time frame was specified to perform the experimental task based on the results of our preliminary study.

6.3.6- Participants

Participants were invited to participate in the experiment based on our inclusion criteria. All engineers involved in the ICS development process were identified as potential participants, such as control engineers, control system designers and embedded system engineers. Engineering students were a representative sample of the ICS-SES tool's intended users. According to the guideline developed by Ko *et al*., "students can be appropriate participants when their knowledge, skills, and experiences fit within a tool's intended user population" (Ko et al., 2015). In the survey reported by (Sjøberg et al., 2005), it was found that students had been recruited for 91 controlled experiments in software engineering.

The experiment used 79 participants in the faculty of Technology at De Montfort University, Leicester, UK. The participants had different levels of education (undergraduates, postgraduates and lecturers). They were randomly assigned to the treatment groups, leading to 40 in the Supported group and 39 in the Plain group.

The sample size was suitable for evaluating our framework for several reasons: (1) any noise and variation between the experimental groups were minimised by conducting the experiment at the same time in the same environment using the same material, and providing the same development environment except for the use, or otherwise, of the ICS-SES tool. The less variation within an experiment, the fewer participants one needs (Ko et al., 2015). (2) a significant difference can be seen across a small group of participants in each experimental group; for example, the study conducted by (LaToza and Myers, 2011) achieved significant differences with just six respondents per group. (3) according to the review of the 92 controlled experiments reported by Dybå *et al*, such a sample size is widely accepted in software engineering (Dybå et al., 2006). Their results showed that the average sample size average is 34 per group.

In addition, the sample size was calculated based on the formula reported by (Allen Jr, 2011) using the SPSS software suite. Since there was no historical data from similar studies available, the effect of size was determined using the two pilot samples' means from the results of our preliminary study. The calculation resulted in a sample size of 70 for α=0.05, power =0.80.

6.4- Experiment Procedure

The experiment procedure was developed based the methods of (Ko et al., 2015) The experimental session was designed to cover the five parts presented in Figure 6.8. Before participants begin the experiment, they were given an informed consent form, which provides a brief explanation of our research and the purpose of the experimental study in order to allow them to decide whether they wanted to participate, as shown in Appendix B-4. The informed consent was approved through the ethics approval process for this study.

The group assignment was done randomly to distribute the random variation in respondents' security backgrounds across the two groups. Random assignment to experimental conditions is widely used in controlled studies to ensure that differences in the groups' performance is due to any differences in conditions or tools being compared, rather than differences between participants (Ko et al., 2015).

A tutorial about the scenario was given to both groups. At the end of the tutorial, the ICS-SES tool was introduced to the experiment group. Then, participants were invited to complete the pre-questionnaire. Once all participants had done so, they were asked to solve the security problem using either the ICS-SES tool or the Plain tool as based on the group they had been [randomly] assigned to. When a participant successfully solved the problem or the time allocated to the task expired, they were invited to complete the appropriate post-questionnaire (again, as based on the group they had been assigned to).

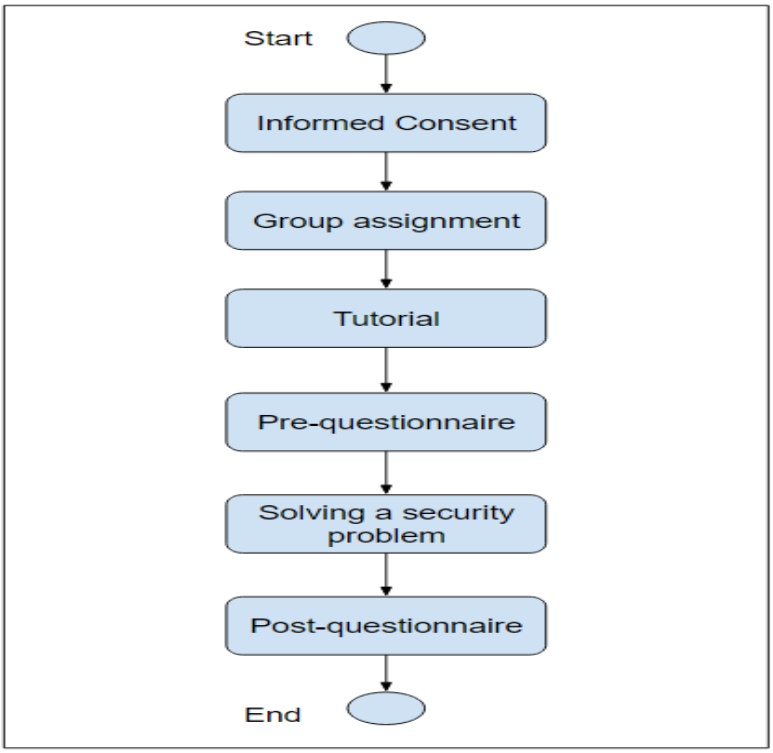


Figure 6-8 The experimental procedure

6.5- Preliminary study

A preliminary study was conducted to test the viability of the experimental procedure and to estimate the time required to complete the experimental session. Eleven engineers voluntarily participated in the experiment. They were randomly assigned to two groups. The Supported group included six participants, one lecturer, four postgraduates and one undergraduate student. The Plain group had five participants, one lecturer, three postgraduates and one undergraduate students.

The experiment was conducted in a computer laboratory in the Faculty of Technology at De Montfort University. Participants went through the whole experimental procedure, including the tutorial, pre-questionnaire, experiment task and post-questionnaire. All participants completed the session within one hour.

6.5.1. Preliminary Results

The initial results helped the researcher to calculate the required sample size for the main controlled experiment, as explained in Section 6.3.6. Participants' feedback also led to the identification, and subsequent removal, of a number of defects, as well

as improving the experimental design before conducting the main evaluation experiment.

The feedback collected showed that the questions in the pre-test and post-test were too specific. Therefore, they were changed to be more general and assess the understanding of the problem and the security solution, including a scenario of a security problem that was similar to the one given in the experimental task (see Appendix D).

The ICS-SES tool was also improved based on the feedback of the Supported group. Participants' feedback showed that some topics in the training material are complex, and some terminologies were hard to understand. Thus, the training content was simplified and any related terminology was defined. In addition, participants commented about the task load of using the tool, as users had to select a security pattern and browse the related training material. Therefore, offering the training material was changed so as to be optional.

Regarding the experiment material, participants were able to understand the experiment instructions, system scenario and the questions in both questionnaires.

6.6- Experiment Execution

The main experiment was conducted in the computer laboratory in the Faculty of Technology at de Montfort University. Participants were randomly assigned to two groups, as shown in Table 6.1. The Supported group had forty participants (twenty-five undergraduate and fifteen postgraduate students). The Plain group had thirty-nine participants (twenty-four undergraduate and fifteen postgraduate students).

Participants	Supported group	Plain group	Total
Undergraduate	25	24	49
Postgraduate	15	15	30
Total	40	39	79

Table 6-1 Participants in the experiment groups

Initially, the researcher presented a familiarisation tutorial about the system scenario and explained the instructions to the experimental process. There was a section in the tutorial for the Supported group to introduce our ICS-SES tool. During the tutorial,

participants were given as much time as they required to ask questions and clarify the scenario.

When all participants in both groups had completed the pre-questionnaire, they were asked to start the task of solving the security problem in the scenario using either the ICS-SES tool or the Plain tool, as based on their assigned group. The Plain group was allowed to use any online materials they liked to help them solve the problem. The time for task completion was limited to half an hour. However, participants were asked to inform the conductor when they successfully solved the problem to record the time. Finally, participants completed the post-questionnaires based on their assigned group and were informally asked to give their feedback and attitude toward the ICS-SES tool, where appropriate.

The data collected was analysed based on its type, categorical or numerical, as presented and discussed in Chapter 7.

6.7- Conclusion

This chapter presented an empirical study design that was used to assess the usability of the ICS-SES educational tool in terms of effectiveness, efficiency and ease of task in assisting engineers to develop ICS security by design. The controlled experiment was designed to test five hypotheses, as outlined in Section 6.2, based on the guidelines developed by (Ko et al., 2015) in line with the research question. The experimental design included one treatment with two conditions: an experimental group, ‘Supported’, that used our tool, ‘ICS-SES’, and a control group, ‘Plain’, that did not use our tool. In this experiment, the task of solving a security problem in a system model using the ICS-SES tool support was evaluated in comparison to the traditional development environment simulated by the Plain tool.

Experimental materials were prepared to present the task scenario and data collection instruments, in addition to a familiarisation tutorial to demonstrate the scenario and introduce the ICS-SES tool to the experimental group.

In this chapter, the experiment procedure was explained and the sample size was discussed and justified. Despite the applicability of the experiment procedure, as confirmed by the preliminary results, the feedback highlighted that there were

opportunities for further enhancements. Accordingly, any required changes were implemented and the experimental design was improved.

In the next chapter, the data collected will be analysed and the results presented and discussed in relation to the study hypotheses.

Chapter 7
ICS-SES Evaluation

Chapter objectives

- To introduce the evaluation of the ICS-SES framework
- To present data analysis and results
- To discuss the internal and external threats to validity
- To discuss the findings

7.1. Introduction

This chapter presents the results of the controlled experiment, which was demonstrated in the last chapter, to evaluate our supported framework, 'ICS-SES'. Data was collected from the pre-questionnaire, post-questionnaire and the record of participants' performances that included task completion and duration.

The collected data was analysed using the SPSS software (SPSS, 2013) using various analytical techniques, according to data type, in line with the purpose of this study as outlined in Chapter 6, Section 6.2. Cross-tabulation analysis and Chi-Squared statistics were used to test this research hypothesis regarding data comparison of participant performance, learning and the ease of task. The numerical data, which was collected from the task completion time record, was analysed by calculating the mean total time for each experimental group, and the two group results were compared using independent two-sample t-test analysis to test our hypothesis regarding efficiency. The subjective feedback was analysed based on the cognitive dimension framework. In addition, the Google Analytics tool was used to analyse participants' behaviour and engagement with the provided training.

The chapter is divided into the following sections. Section 7.2 presents the results of the evaluation experiment in relation to the dependent variables. Section 7.3 presents the subjective feedback collected by the Cognitive Dimension (CD) framework and

overall users' experiences. Section 7.4 presents the results of participants' engagement with the security training. Section 7.5 illustrates the Plain group's experience. Section 7.6 highlights the internal and external threats. Section 7.7 discusses the experimental results in relation to the study hypotheses. Section 7.8 summarises the chapter.

7.2. Results

A dataset with 79 responses collected from the pre-questionnaire, 79 responses from the post-questionnaire and 79 responses from participants' performance record was produced. The data was entered into the SPSS software suite to be analysed using a number of analytical techniques.

The results are presented and grouped according to the objectives of our research.

7.2.1. Participants' Prior knowledge

7.2.1.1. ICS Security Problems

"How do you rate your knowledge about common security problems in industrial control systems?"

The result shows that a large proportion of the responses (74%) rated their ICS security issues knowledge as being at a poor, or very poor, level. A small proportion (18%) of the respondents rated their knowledge as average, while only 8% reported that their understanding of ICS security is good. None of the participants rated their knowledge as an excellent, as shown in Figure 7.1. The results clearly show that engineers lack knowledge of ICS security problems. Our automated tailored training tool should be useful in enhancing engineers' security knowledge.

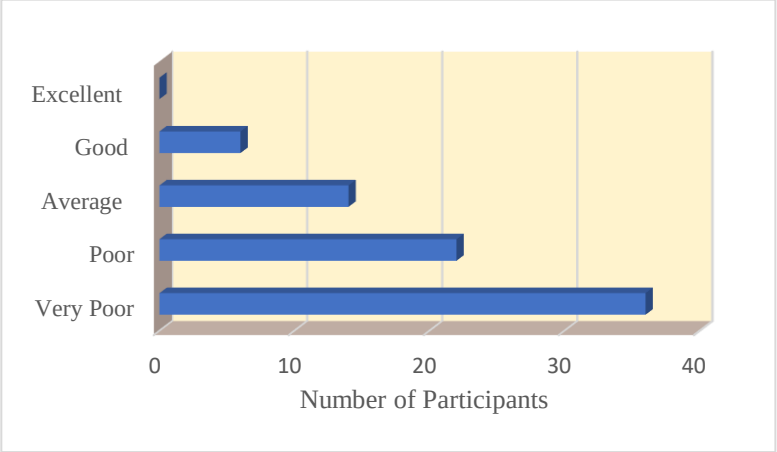


Figure 7-1 Participants' knowledge on ICS security problems

7.2.1.2. Security Standards and Guidelines

"How do you rate your knowledge about security standards and guidelines for industrial control system design?"

Figure 7.2 shows that 81% of participants rated their knowledge of ICS security standards as being at a poor, or very poor, level, whereas a small minority (13%) reported their background as being average. The remaining minority (6%) claimed a good level of knowledge. None of the responses rated themselves as having an excellent level of knowledge. The results illustrate that engineers lack knowledge of security standards and best practise.

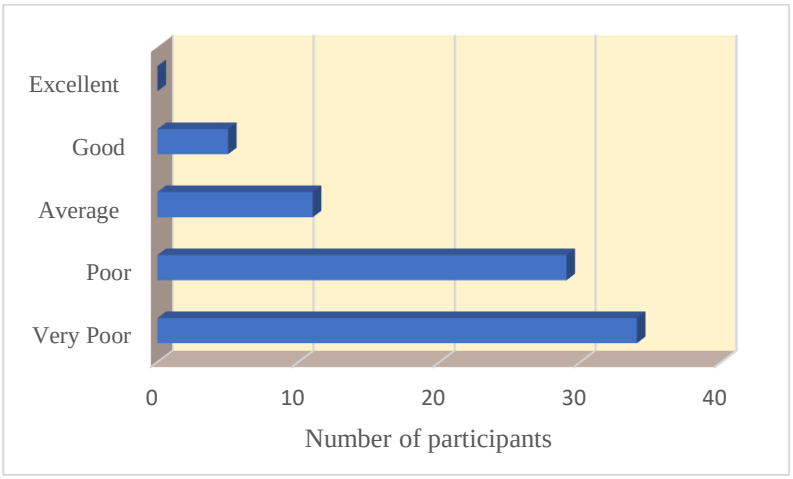


Figure 7-2 Participants' knowledge on ICS security standards

In terms of investigating security awareness, Figure 7.3 presents the results of participants' awareness of the common institutions and teams that publish ICS security guidelines and recommendations. The bar chart below shows that almost all participants had not heard about those publishers, with 13% of participants having heard about them but not used them. The responses show that NIST has been used by 1.4% of the participants, followed by 0.3% who have used SANS and 0.2% who have used NISA. The results demonstrate that ICS developers are not aware of common security guidelines' resources, therefore the tool should be useful in providing a repository of knowledge that engineers can easily access.

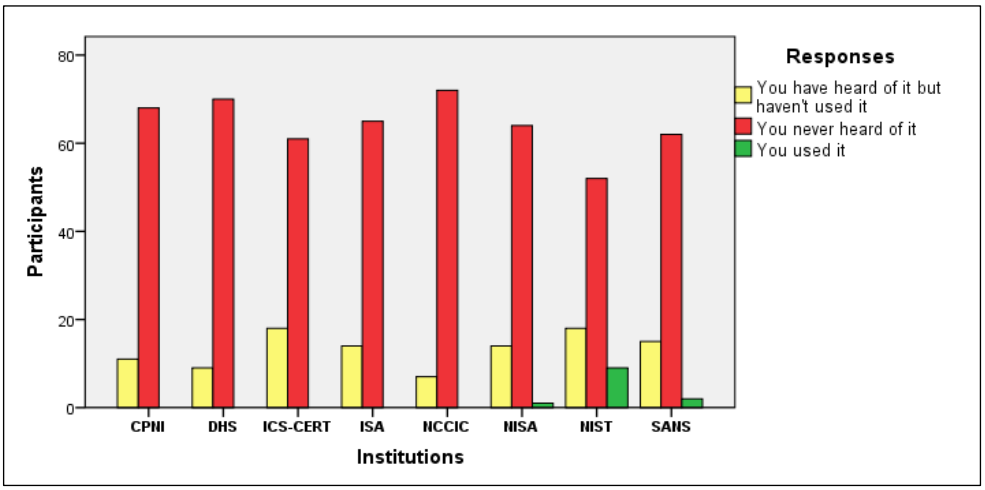


Figure 7-3 Participants' awareness of common security guideline publishers

7.2.1.3. Security Engineering Awareness and Responsibility

"*Are you responsible for the security in the control system development?*"

Figure 7.4 presents the responses for participants' awareness of the need to consider security early in the system development cycle. Just over a half (53%) of participants stated that they are not responsible for secure control systems, whereas 42% reported that they share the role of security engineering. Only 5% of the participants claimed full responsibility for building secure systems. The results show that control system engineers are not aware of their responsibilities for developing secure systems. As the ICS-SES tool is integrated into everyday work, it should be useful in supporting engineers' roles in security engineering.

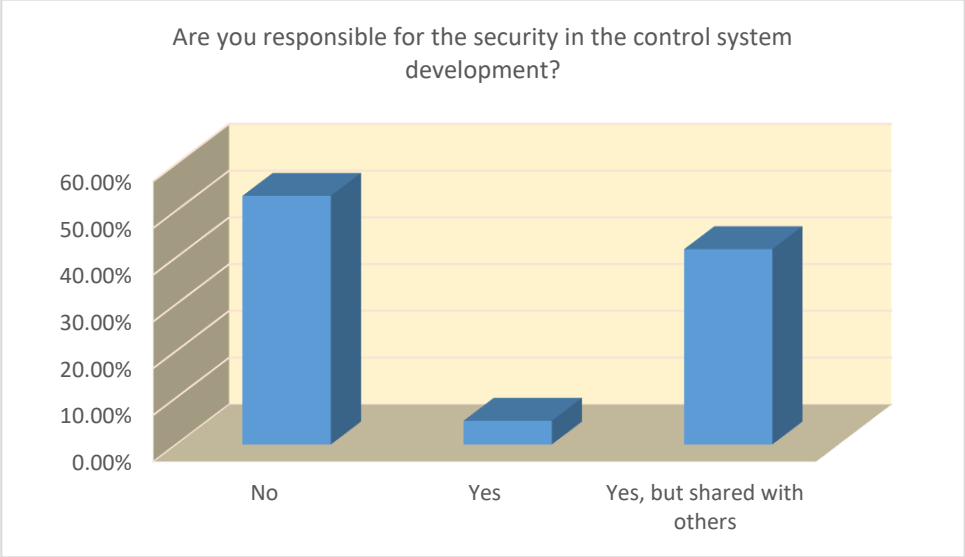


Figure 7-4 Participants' awareness of the security engineering responsibility

"When you design a control system, do you take security requirements into consideration?"

Similarly, regarding security engineering, these responses articulate the lack of security consideration during the system design phase. A significant majority (83%) of the participants have never considered system security during development cycle. 10% rarely take the security requirements into account. 5% of responses showed some security consideration whilst an almost insignificant number (2%) of participants declared that they always consider security during system design, as shown in Figure 7.5. The results demonstrate that control system development process lack security requirements consideration. Therefore, the ICS-SES tool would bring security into process.

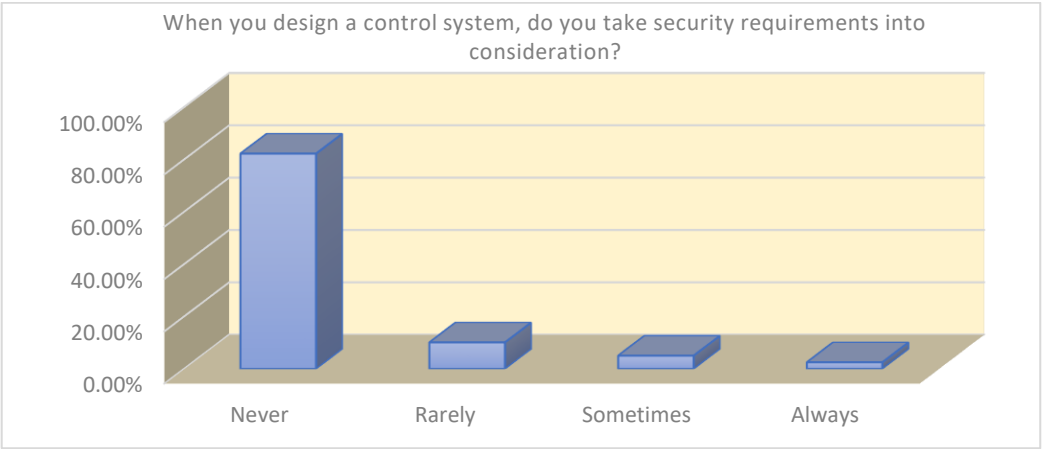


Figure 7-5 Security requirements consideration during system design

"At which phase of control system development cycle security should be considered?"

The pie chart below presents the responses for the awareness of considering security throughout the system lifecycle. In contrast to the previous result, almost half of participants (48%) stated that security should be considered at all system development phases. A small minority (17%) stated that security should be considered at the design phase, with the remainder reporting consideration at the operation and building phase, in the proportions of 13% and 6%, respectively. 16% of participants did not know at which phase security should be considered, as shown in Figure 7.6. The results show that engineers lack security awareness, therefore our educational tool should be useful in increasing engineers' security awareness.

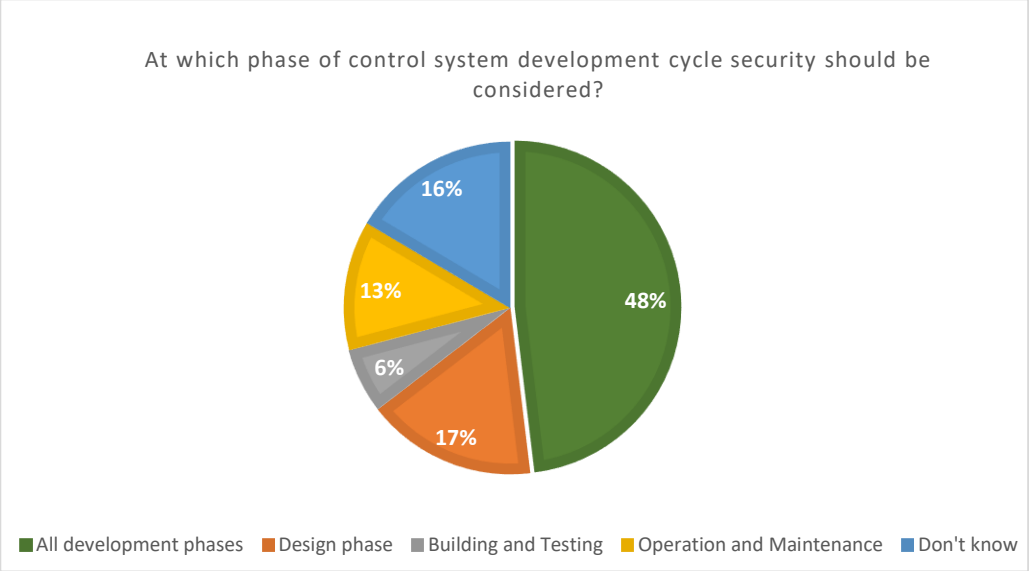


Figure 7-6 Responses as to the phase at which system development should involve security considerations

To sum up, the previous results were grouped together because they demonstrate the participants' current awareness and knowledge. The results have shown that the level of understanding amongst control system engineers is not what is needed to produce a secure system.

7.2.2. Security Training

"Have you had any training on control system security before?"

Almost all participants (95%) have not had any security training, as shown in Figure 7.7. Only four participants out of 79 had training courses and it was more than five years ago, 3 of them had security courses as a part of their education and one at work place. The results clearly demonstrate that control system developers lack security training support, our tool should be useful as it provides on-the-job training support.

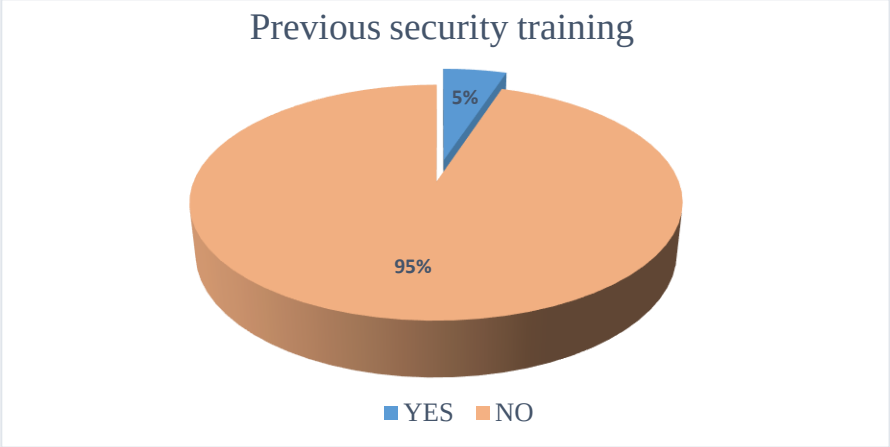


Figure 7-7 Participants' responses to attending previous security training

7.2.3. Engineers' Motivation

"When a security weakness is discovered during system development, what does describe your most common action?"

Figure 7.8 shows that, a significant majority of participants (74%) expressed an interest in learning about security problems and how to solve them. Approximately a quarter of the participants (24%) would pass the problem to different team. Only one participant expressed a complete disregard for the security problem. The results show that the ICS-SES tool should be useful as it fulfils the learning needs of control system engineers.

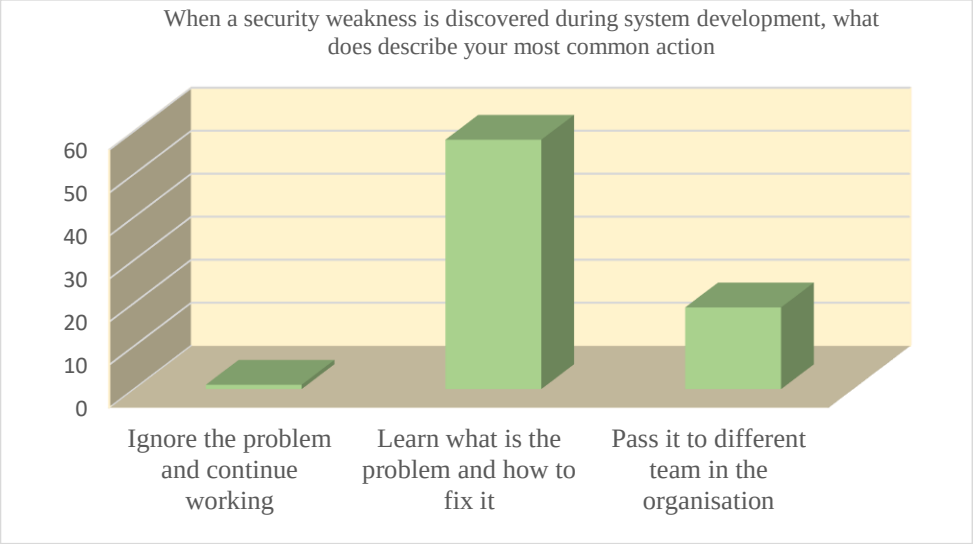


Figure 7-8 Participants' reaction of finding a security problem during system development

"In terms of learning new skills, what is your preferred training method?"

In terms of investigating the motivation of the participants, they were asked to identify the preferred training method. The responses showed that more than half (58%) preferred to learn during their work, while a quarter (25%) liked to learn through regular courses. A small minority of participants (15%) stated a preference for discussion and workshop methods, and one of participant suggested online tutorials for learning new skills. This data is reported in Figure 7.9. The results show that the right design has been chosen for the supported tool, as it will fit the need in the manner that the users would prefer.

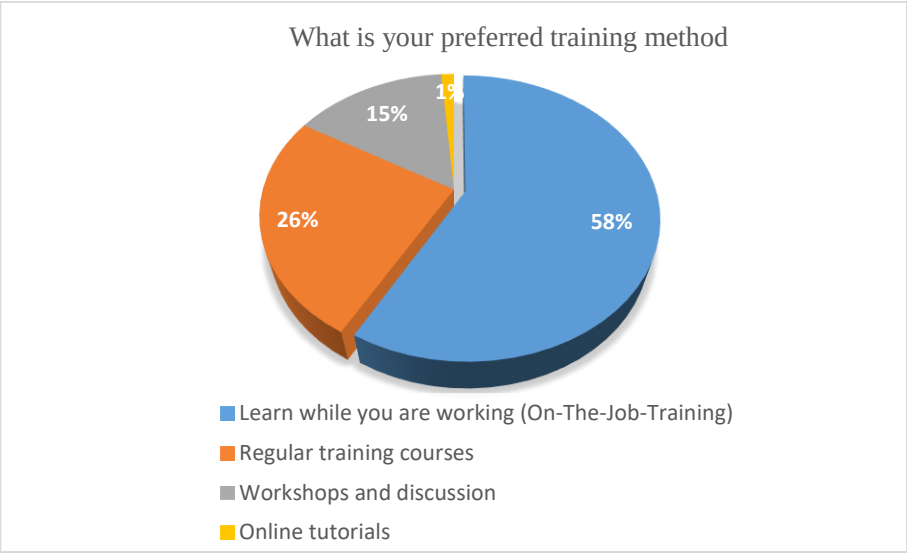


Figure 7-9 Participants' responses as to preferred training methods

7.2.4. Results of Comparison

7.2.4.1. Results of effectiveness (successfully solved the security issue)

Since the results of the evaluation experiment were categorical data, cross-tabulation analysis was used for data analysis. The task completion results of 79 participants were entered into SPSS. The performance of the two groups was compared using cross-tabulation, where the two groups were set as rows and the results of the task performance were set as a column, as shown in Table 7.1. Figure 7.10 illustrates the resulting comparison between the two groups based on the cross-tabulation analysis. The results show that 95.0% of participants from the Supported group successfully solved the security problem given in the experimental task, which is 77.1% higher than the Plain group. All participants in the Supported group reported that they performed the task with the help of our tool.

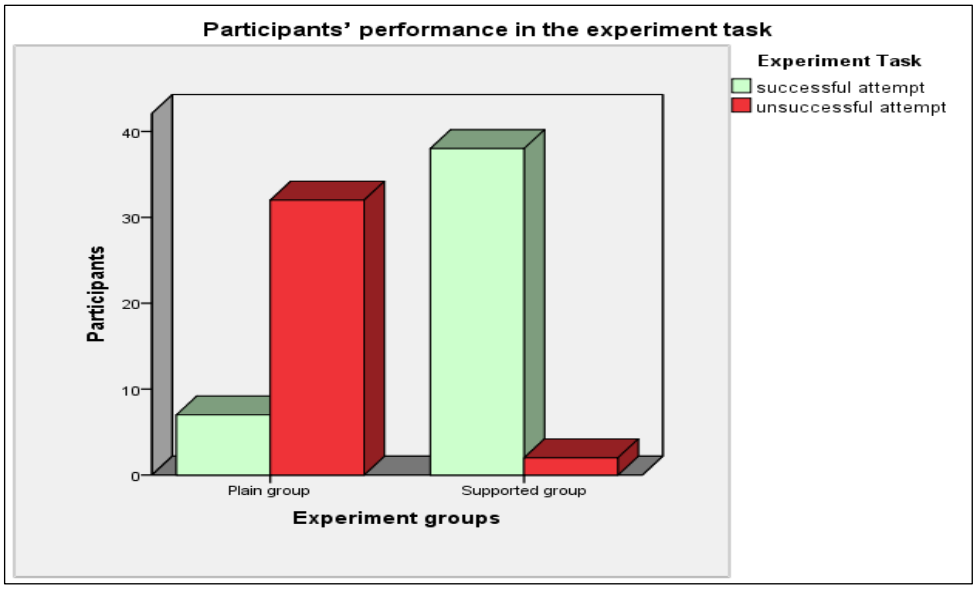


Figure 7-10 Comparing the performance of the experimental task between Supported and Plain group

		Experiment task		Total
		Successful attempts	Unsuccessful attempts	
Plain group	Count	7	32	39
	% of Total	17.9%	82.1%	100.0%
Supported group	Count	38	2	40
	% of Total	95.0%	5.0%	100.0%
Total	Count	45	34	79
	% of Total	57.0%	43.0%	100.0%

Table 7-1 The performance of the experimental task using cross-tabulation

Chi-Squared statistics was also used in order to test the hypothesis, 'H_1', which was identified in Section 6.2. The Chi-Squared test ($P < 0.001$) shows that the results are statically significant, as shown in Table 7.2.

	Chi-Squared Tests		
	Value	*df*	*Asymptotic Significance (2-sided)*
Pearson Chi-Squared	47.821[a]	1	.000
Continuity Correction	44.730	1	.000
Likelihood Ratio	55.391	1	.000
N of Valid Cases	79		

Table 7-2 Chi-Squared Test performance of the experimental task using cross-tabulation

7.2.4.2. Results of effectiveness (Learning outcomes)

The results of the pre-test and post-test were separately analysed and compared using cross-tabulation analysis for both the security problem and solution.

Understanding the problem

Table 7.3 compares the pre-test results of the two groups regarding problem understanding. The results show that an insignificant amount of responses were correct, which were 7.5% and 5.1% for the Supported and Plain groups, respectively. The total percentage of participants who did not know the answer and answered incorrectly was 92.5% in the Supported group, which is almost the same as the Plain group. The Chi-Squared test ($P > 0.05$) also shows that the results were statically insignificant, as shown in Table 7.4.

		Answers to Pre-test			Total
		Correct	Don't know	Incorrect	
Plain group	Count	2	15	22	39
	% of Total	5.1%	38.5%	56.4%	100.0%
Supported group	Count	3	21	16	40
	% of Total	7.5%	52.5%	40.0%	100.0%
Total	Count	5	36	38	79
	% of Total	6.3%	45.6%	48.1%	100.0%

Table 7-3 Pre-test results for problem understanding using cross-tabulation

Chi-Squared Tests			
	Value	***df***	***Asymptotic Significance (2-sided)***
Pearson Chi-Square	2.135[a]	2	.344
Likelihood Ratio	2.145	2	.342
N of Valid Cases	79		

Table 7-4 Chi-Square tests pre-test results for problem understanding using Cross-tabulation

In terms of analysing the post-test results, the same analysis process as used for the pre-test results was followed. Table 7.5 illustrates that almost all participants in the Supported group achieved correct answers (92%), which is 66.9% higher than for the Plain group. As presented in Table 7.6, the Chi-Square test ($P < 0.001$) shows that the results of the post-test of security problem are statically significant.

		Answers of Post-test			Total
		Correct	Don't know	Incorrect	
Plain group	Count	10	2	27	39
	% of Total	25.6%	5.1%	69.2%	100.0 %
Supported group	Count	37	2	1	40
	% of Total	92.5%	5.0%	2.5%	100.0 %
Total	Count	47	4	28	79
	% of Total	59.5%	5.1%	35.4%	100.0 %

Table 7-5 Post-test results for problem understanding using Cross-tabulation

Chi-Square Tests			
	Value	***df***	***Asymptotic Significance (2-sided)***
Pearson Chi-Square	39.647 [a]	2	.000
Likelihood Ratio	46.677	2	.000
N of Valid Cases	79		

Table 7-6 Chi-Square Tests Post-test results for problem understanding using Cross-tabulation

Understanding the solution

Table 7.7 shows the results of testing pre-existing knowledge of the security solution by comparing the two groups. Only a small minority of both groups' responses were correct, which was around 5%, while 95% of the responses in each group were not correct. As presented in Table 7.8, The Chi-Squared test ($P > 0.05$) shows that the test results regarding prior knowledge of the security solution are statically insignificant.

		Answers			Total
		Correct	Don't know	Incorrect	
Plain group	Count	2	18	19	39
	% of Total	5.1%	46.2%	48.7%	100.0%
Supported group	Count	2	19	19	40
	% of Total	5.0%	47.5%	47.5%	100.0%
Total	Count	4	37	38	79
	% of Total	5.1%	46.8%	48.1%	100.0%

Table 7-7 Pre-test results for solution understanding using cross-tabulation

Chi-Squared Tests			
	Value	***df***	***Asymptotic Significance (2-sided)***
Pearson Chi-Square	.014[a]	2	.993
Likelihood Ratio	.014	2	.993
N of Valid Cases	79		

Table 7-8 Chi-Squared test pre-test results for solution understanding using cross-tabulation

In contrast, as presented in Table 7.9, the comparison between the two groups' responses shows that 85% of the results of post-test on the security solution were correct in Supported group, which was 74.7% higher than for the Plain group. The Chi-Squared test ($P < 0.001$) shows that the results of the security solution post-test are statically significant, as shown in Table 7.10.

		Answers			Total
		Correct	Don't know	Incorrect	
Plain group	Count	4	13	22	39
	% of Total	10.3%	33.3%	56.4%	100.0%
Supported group	Count	34	3	3	40
	% of Total	85.0%	7.5%	7.5%	100.0%
Total	Count	38	16	25	79
	% of Total	48.1%	20.3%	31.6%	100.0%

Table 7-9 Post-test results for solution understanding using cross-tabulation

Chi-Square Tests			
	Value	***df***	***Asymptotic Significance (2-sided)***
Pearson Chi-Square	44.369[a]	2	.000
Likelihood Ratio	50.142	2	.000
N of Valid Cases	79		

Table 7-10 Chi-Squared test post-test results for solution understanding using cross-tabulation

The results show that participants in both groups have similar prior knowledge on the problem context and corresponding solution. In contrast, the learning outcomes of Supported group were significantly better than Plain group.

7.2.4.3. Results of efficiency (time)

The task durations for the 79 participants were recorded and analysed. Table 7.11 shows that the mean total time spent by the Supported group (N = 40) was M = 13.65 minutes (SD = 3.86). By comparison, the mean time taken by the Plain group (N = 39) was significantly greater at M = 29.3 minutes (SD = 1.49). Since the results are numerical data, independent two-sample t-test analysis was used to test the hypothesis, 'H_4', that the mean time of the task duration taken by the two groups was statistically significantly different. As can be seen in Table 7.12, there was a significant difference in participants' efficiency in performing the experiment task in the Supported group (M = 13.65, SD = 3.86) and the Plain group (M = 29.3, SD = 1.49); (t (77) = 23.73, P = 0.000 < 0.001). The results show that Supported group took less time than Plain group.

	Experiment group	N	Mean	Std. Deviation	Std. Error Mean
Task duration	Supported group	40	13.6500	3.86669	.61138
	Plain group	39	29.3846	1.49764	.23981

Table 7-11 The mean total time taken to complete the experiment task

	Levene's Test for Equality of Variances		t-test for Equality of Means					
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference 95% Confidence Interval of the Difference	
							Lower	Upper
Equal variances assumed	10.934	.001	23.733	77	.000	15.73462	14.41445	17.05478
Equal variances not assumed			23.959	50.693	.000	15.73462	14.41598	17.05325

Table 7-12 T-test results for Supported and Plain group efficiency in the experiment task

7.2.4.4. Results of ease of task (solving the problem)

The results were analysed in order to compare the difficulties during performing the task, which involved understanding the security problem, and finding a solution and understanding it, between the two groups.

Figure 7.11 shows that in the Supported group, a majority of participants (77.5%) stated that they had no difficulties in understanding the security problem. Only 5% of participants stated that they had had some difficulties. In contrast, a majority of

the Plain group (71.8%) reported that they had difficulty in understanding the problem.

Table 7.13 shows that the Supported group demonstrated a significantly less level of difficulty in understanding the security problem than Plain group. The Chi-Squared test ($P < 0.001$) also shows that the responses to difficulty in understanding the security problem are statically significant, as shown in Table 7.14.

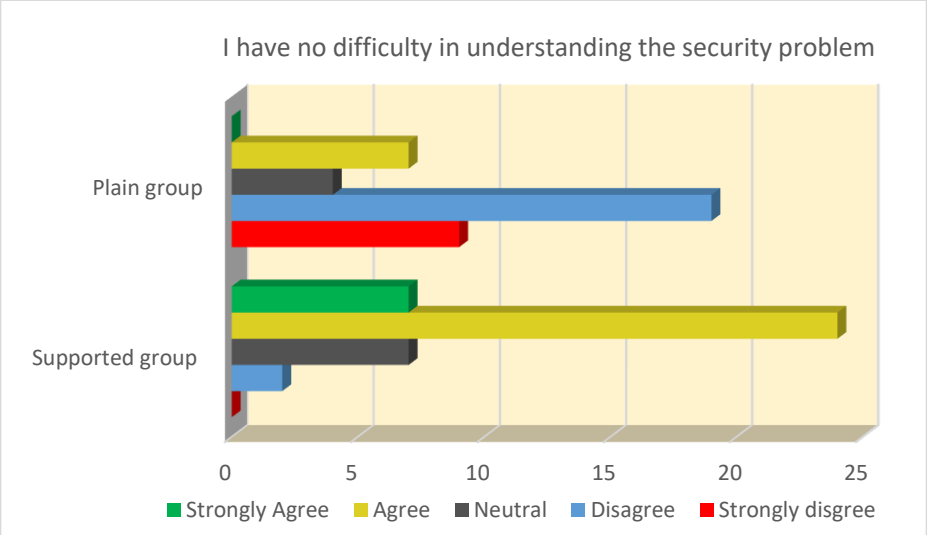


Figure 7-11 Participants' responses to having difficulty in understanding the security problem

		No difficulty in understanding the problem					Total
		Agree	*Disagree*	*Neutral*	*Strongly Agree*	*Strongly Disagree*	
Plain group	Count	7	19	4	0	9	39
	% of Total	17.9 %	48.7%	10.3%	0.0%	23.1%	100.0 %
Supported group	Count	24	2	7	7	0	40
	% of Total	60.0 %	5.0%	17.5%	17.5%	0.0%	100.0 %
Total	Count	31	21	11	7	9	79
	% of Total	39.2 %	26.6%	13.9%	8.9%	11.4%	100.0 %

Table 7-13 Participants' responses to understanding the security problem using cross-tabulation

Chi-Square Tests			
	Value	*df*	*Asymptotic Significance (2-sided)*
Pearson Chi-Square	39.896[a]	4	.000
Likelihood Ratio	48.757	4	.000
N of Valid Cases	79		

Table 7-14 Chi-Squared tests results of difficulty in understanding the security problem using cross-tabulation

Figure 7.12 shows that a large proportion of participants from the Supported group (62.5%) had no difficulties in identifying a security solution for the problem. However, 10% of the responses expressed some difficulties. The majority of the Plain group (82.1%) experienced difficulty in finding a possible solution. Only two out of thirty-nine of the participants stated that they had had no difficulty in identifying the solution.

As can be seen in Table 7.15, the responses of the Supported group demonstrated less difficulty in finding a solution to solve the security problem than the Plain group. Table 7.16 shows the difference between the two groups' results using the Chi-Squared test ($P < 0.001$), which shows that the results are statically significant.

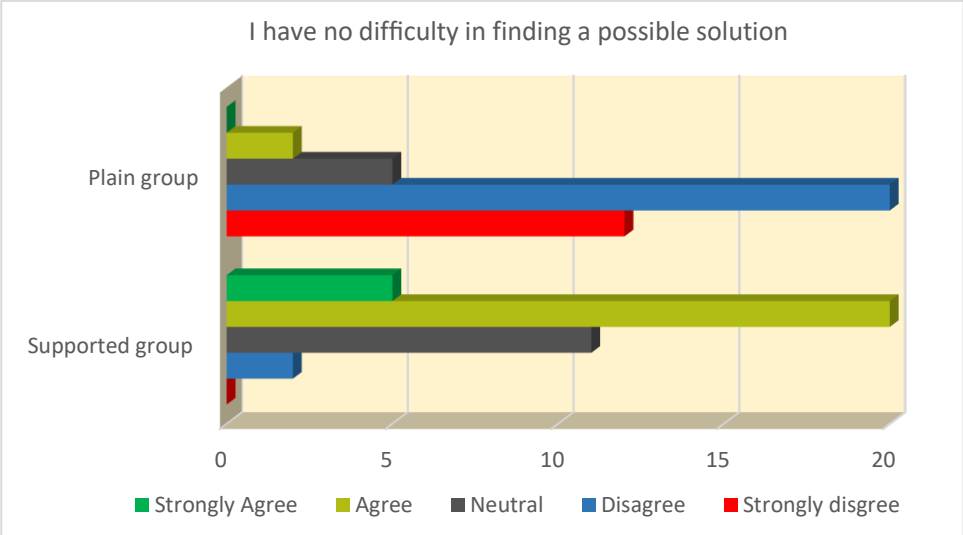


Figure 7-12 Participants' responses to having difficulty finding a security solution

		No difficulty in finding a possible solution					Total
		Agree	*Disagree*	*Neutral*	*Strongly Agree*	*Strongly Disagree*	
Plain group	Count	2	20	5	0	12	39
	% of Total	5.1%	51.3%	12.8%	0.0%	30.8%	100.0%
Supported group	Count	20	4	11	5	0	40
	% of Total	50.0 %	10.0%	27.5%	12.5%	0.0%	100.0%
Total	Count	22	24	16	5	12	79
	% of Total	27.8 %	30.4%	20.3%	6.3%	15.2%	100.0%

Table 7-15 Participants' responses to finding a possible solution

Chi-Square Tests			
	Value	df	Asymptotic Significance (2-sided)
Pearson Chi-Square	44.638[a]	4	.000
Likelihood Ratio	54.599	4	.000
N of Valid Cases	79		

Table 7-16 Chi-Squared test results for difficulty in finding a possible solution

Regarding the difficulty in understanding the security solution, 29 out of 40 of the Supported group reported that they had had no difficulty compared to one participant from the Plain group, as shown in Figure 7.13.

Table 7.17 shows the differences between the groups. The total percentage of the Supported group of having difficulties in understanding the solution was 5%, which is 76.6% less than for the Plain group. According to the Chi-Squared test ($P < 0.001$), those results are statically significant, as shown in Table 7.18.

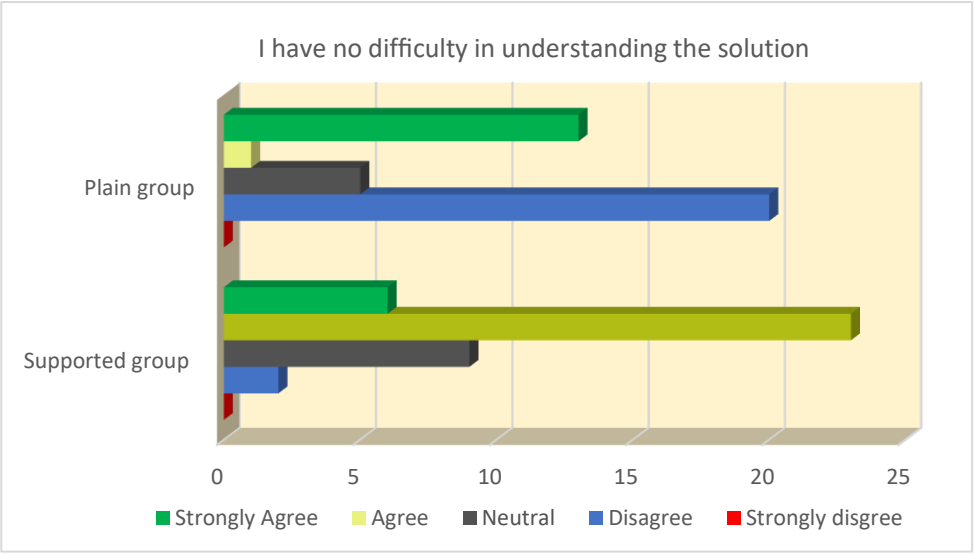


Figure 7-13 Participants' responses to having difficulty understanding the security solution

		No difficulty in understanding the solution					Total
		Agree	*Disagree*	*Neutral*	*Strongly Agree*	*Strongly Disagree*	
Plain group	Count	1	20	5	0	13	39
	% of Total	2.6%	51.3%	12.8%	0.0%	33.3%	100.0%
Supported group	Count	23	2	9	6	0	40
	% of Total	57.5%	5.0%	22.5%	15.0%	0.0%	100.0%
Total	Count	24	22	14	6	13	79
	% of Total	30.4%	27.8%	17.7%	7.6%	16.5%	100.0%

Table 7-17 Participants' responses to understanding the solution using cross-tabulation

Chi-Square Tests			
	Value	Df	Asymptotic Significance (2-sided)
Pearson Chi-Square	55.033[a]	4	.000
Likelihood Ratio	69.538	4	.000
N of Valid Cases	79		

Table 7-18 Chi-Squared test results for difficulty in understanding the solution using cross-tabulation

7.3. Subjective Feedback

This section analyses the subjective data collected from the Supported group. The feedback was collected regarding the usability and usefulness of ICS-SES. The participants were given the option to use our tool to perform the experiment task. Their responses demonstrated that all participants performed the task with the help of our tool.

7.3.1. Evaluation using Cognitive Dimension (CD)

As discussed in Section 6.3.3.5, the evaluation of ICS-SES's usability was based on the Cognitive Dimension (CD) analytical framework. The six dimensions were analysed, as shown in Figure 7.14.

CD 1- Visibility Dimension

32 out of 40 of the participants agreed or strongly agreed that "The tool allows access to all of the relevant information easily". There was one participant who disagreed with this sentiment, who commented that "*Some topics have too much text to read*". The suggestions were "*consider using different colours and more diagrams*". Since too little information will make it difficult to understand and a large portion of the participants were pleased with the training material, we decided to not make any changes to the training content.

CD 2- Hard Mental Operations Dimension

27 out of 40 of the participants agreed or strongly agreed that "The tool aided in solving hard or complex problems that would not have been possible in my head". One participant disagreed and argued that "*the provided training needs more time to be understood, it uses difficult terminology*". After preliminary evaluation, which was presented in the last chapter, the training material was already improved by providing definitions for all security terms and similar jargon.

CD 3- Diffuseness Dimension

The majority of the participants, 33 out of 40, agreed or strongly agreed that "The training material provided the full range of information required to solve the problem". Two out of 40 of the participants disagreed, and stated that "*more details needed in training topics"* and *"It didn't go in deeper details*". As too much information in the training material will undermine the effectiveness of the adaptive ICS-SES tool, we decided to not make any changes regarding these comments.

CD 4- Closeness of mapping Dimension

31 out of 40 of the participants agreed or strongly agreed that "The tool accurately portrays the situation in a context that engineers are familiar with". There was one participant who disagreed with this statement, who commented that "*the issue was not very complex, I'm not sure if it works with complex systems*". The participant also suggested providing more options to solve the problem. This suggestion needs to be considered in the future work in order to extend the security patterns catalogue and identify more interrelations and dependencies between the secure design patterns.

CD 5- Consistency Dimension

Three-quarters of the participants agreed or strongly agreed that "The information provided is consistent across topics". One participant disagreed with this, though without providing any further comments or suggestions related to this dimension.

CD 6- Role Expressiveness Dimension

32 out of 40 of the participants agreed or strongly agreed that "The tool allows me to understand why security vulnerabilities occur within engineering designs". Three participants disagreed with this, with the associated feedback being "*it was hard to know whether training topics are related to the problem or solutions*". This recommendation has been accepted by including brief descriptions to introduce the material at the beginning of each topic before providing further technical explanations.

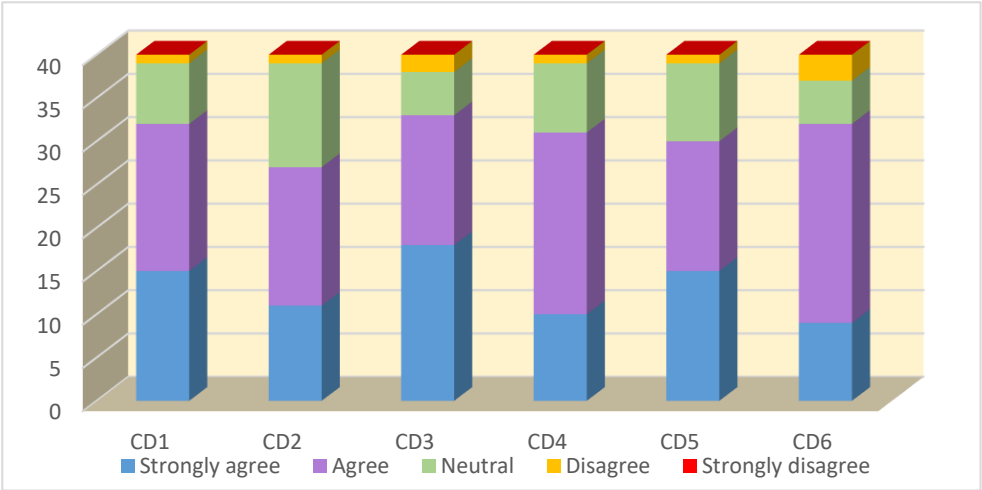


Figure 7-14 The results of ICS-SES usability evaluation using the Cognitive Dimension (CD)

CD1: The tool allows me to access all of the relevant information easily, CD2: The tool aided in solving hard or complex problems that would not have been possible in my head, CD3: The training material provided the full range of information required to solve the problem, CD4: The tool accurately portrays the situation in a context engineers are familiar with, CD5: The information provided is consistent across topics, CD6: The tool allows me to understand why security vulnerabilities occur within engineering designs.

7.3.2. Usefulness and Satisfaction

Figure 7.15 shows the results demonstrating the utility of our tool. Almost all participants agreed or strongly agreed that the tool was useful and helped to understand the problem and the solution. The majority of participants were pleased with the ease of use of the tool. They also expressed the utility of the personal training material.

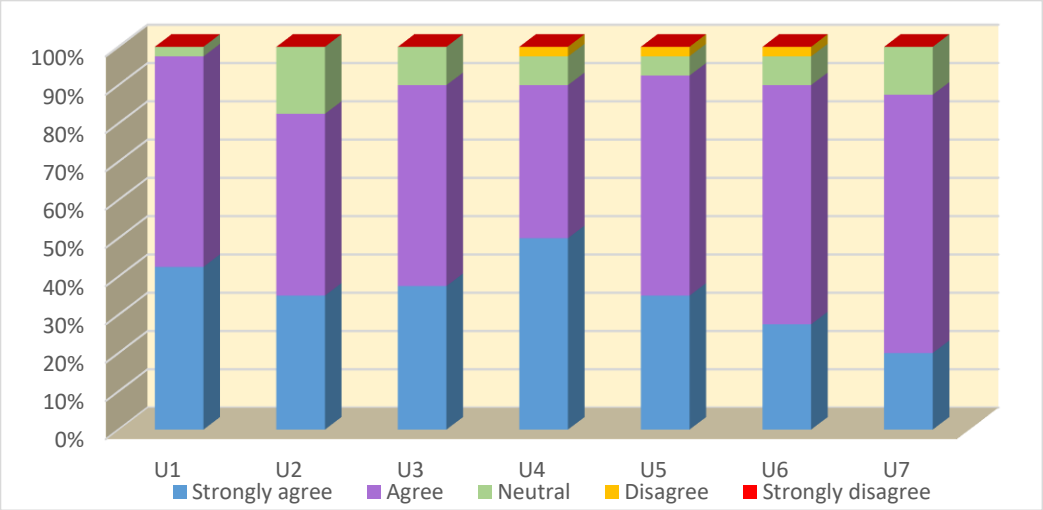


Figure 7-15 The results of ICS-SES usefulness

U1: Overall it is useful, U2: It helps me to understand the problem, U3: It helps me to understand how to solve the problem, U4: The tool is easy to use, U5: The training materials meet my personal needs to understand related security topics, U6: The training material is easy to understand, U7: The tool can help developers in designing secure control systems

Figure 7.16 shows the results associated with user satisfaction. Most participants expressed their satisfaction with the tool and the support training material. Approximately half of participants agreed or strongly agreed that they needed to use the tool to design more secure systems. A large proportion of the participants agreed or strongly agreed that they felt they could use the tool every day at work without undue distraction.

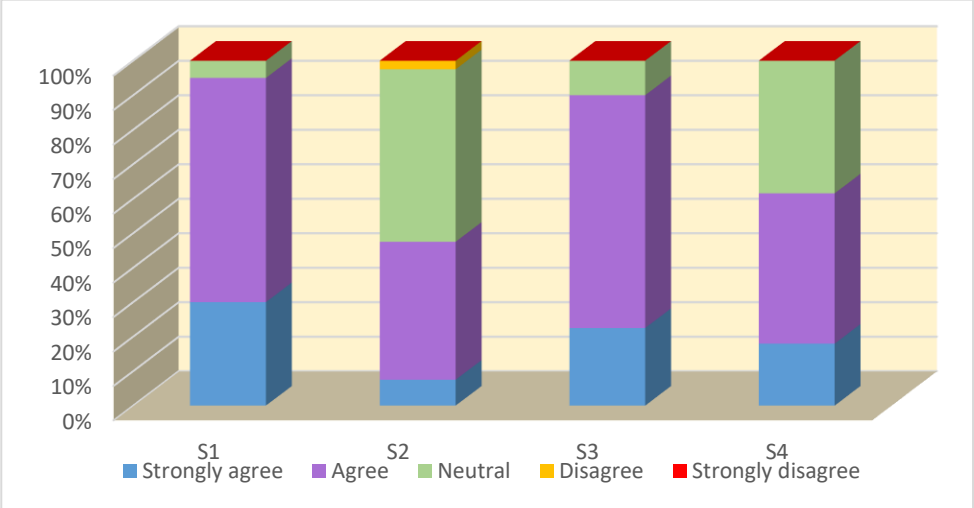


Figure 7-16 The results of ICS-SES user satisfaction

S1: I am satisfied with it, S2: I need to use it to design more secure systems, S3: I am satisfied with the support training material, S4: I can use it easily every day at work without undue distraction.

7.4. Participants' engagement with security training

This section presents the results of the Google Analytics tool that was used to analyse the user engagement with the training material. The researcher created a Google Analytics account, and inserted the associated tracking code into the training sites that presented the training content to be monitored. Participants engagement data was collected and quantitively analysed by Google Analytics. Google Analytics automatically generates a range of reports that can be accessed via the account webpage (Hasan et al., 2009). Only results that provided an overview of user traffic and engagement were considered in terms of monitoring their training activities. The results related to the training topics including five main pages were extracted, as shown in Table 7.19. The results illustrate that all participants viewed the vulnerability training topic. 33 out of 40 of the participants viewed the Access Control training page. The less-viewed training pages were those of Authentication and Authorisation, which were viewed by 22 and 17 of the participants, respectively. 36 out of 40 of the participants viewed the Role-Based Access Control training page.

Training Pages	*Unique page views*
Unauthorised access vulnerability	40
Access control	33
Authentication	22
Authorisation	17
Role-Based Access Control	36

Table 7-19 The results of user engagement with training material

7.5. Plain group experience

The results of the Plain group demonstrated that, overall, participants experienced difficulties in performing the tasks, mostly in understanding the problem and finding and understanding appropriate solutions. One participant described the greatest difficulty as that of being "unfamiliar with terminology".

Figure 7.17 shows that 80% of participants stated that there was some difficulty in finding related information about the problem and how to solve it. 82.5% of the participants disagreed or strongly disagreed with this: "I found required information that meet my personal needs to understand related security topics". Only one out of 39 of the participants agreed with this statement.

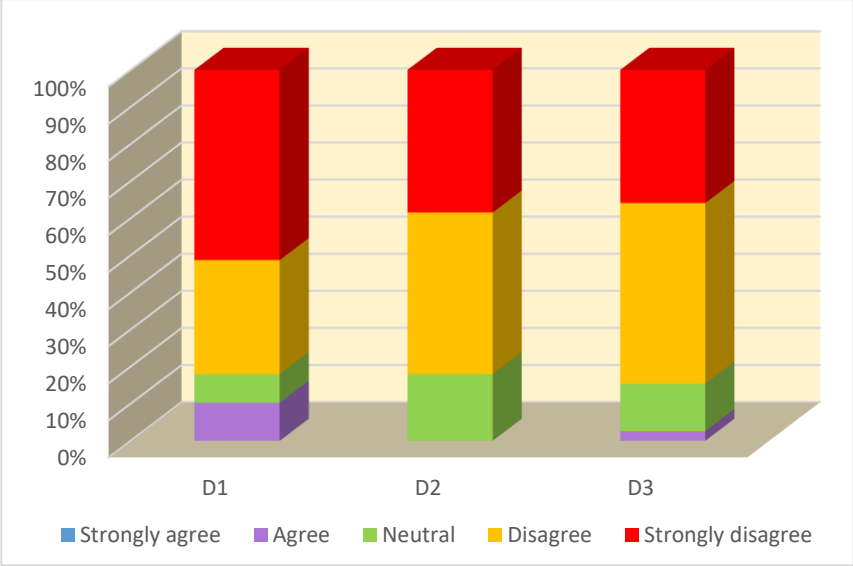


Figure 7-17 Plain group experience on performing the experiment task.

D1: I easily found related information about the problem, D2: I easily found related information about how to solve the problem, D3: I found required information that meet my personal needs to understand related security topics.

As it can be seen in Figure 7.18, 90% of participants expressed their need to understand the problem and the need for guidance with security design. 2.5% of responses stated that there is no need for security understanding or guidance. A large proportion of the participants (77.5%) agreed or strongly agreed that they needed to improve their security knowledge; 10% of participants disagreed with the need to improve such knowledge. Almost all participants agreed or strongly agreed that they needed personal training material that meet their needs; only 2.5% of the participants disagreed. 82.5% of the participants agreed or strongly agreed that they needed greater support in designing secure systems; 2.5% of the participants disagreed.

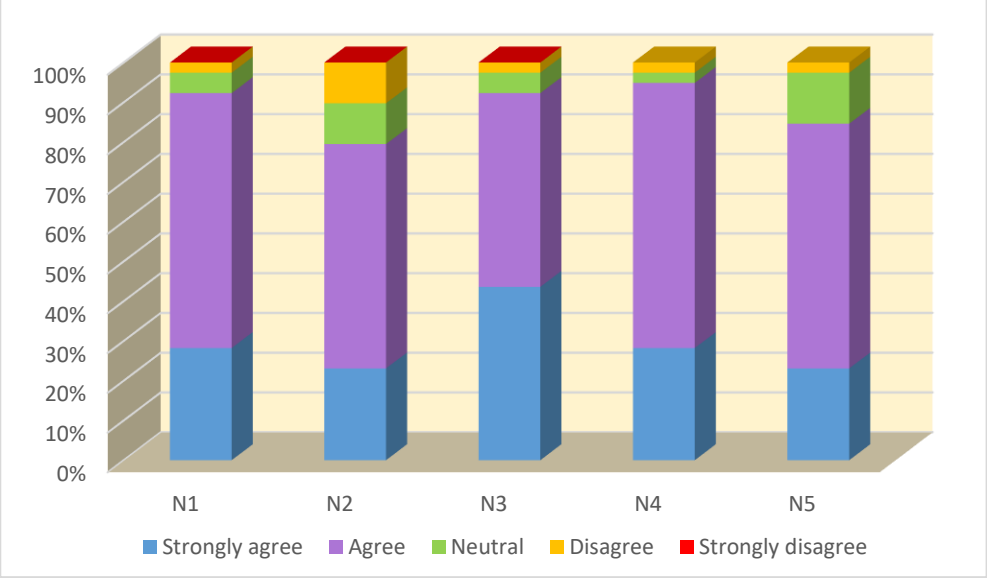


Figure 7-18 Plain group's need to solve the problem.

N1: I need to understand the problem, N2: I need to improve my security skills, N3: I need guidance to solve the security problem in the system model, N4: I need training material that meets my needs in the problem context, N5: I need more support in designing secure systems.

7.6. Internal and External Threats

This section discusses potential threats that could influence the dependent variables other than the chosen independent variables, and illustrates internal and external validity.

7.6.1. Internal Validity

The internal validity of an experiment is "*the validity of inferences about whether observed co-variation between A (the presumed treatment) and B (the presumed outcome) reflects a causal relationship from A to B as those variables were manipulated or measured*" (Shadish et al., 2002).

In this section, internal validity is discussed in relation to the cause-effect relationships induced by this experiment. ***Selection effects***, there was no selection effect in this experiment as the participants were randomly assigned to the two groups. In addition, the results of the pre-test showed that each of the groups had almost the same background. ***Maturity effects***, the boredom threat was controlled by having only one experimental task and limiting its duration to half an hour. ***Learning effects***, there was no learning effect in our experiment because there was only one treatment. We did not use cross-over trials (Stoner, 2004) in the experiment to avoid any learning effect. ***Testing effects***, all participants attended the same familiarisation tutorial and all were given the same material except the treatment. There might be cheating because the participants were in the same place; however, we controlled for this by having two tutors monitoring the participants at all times and by asking them to work independently. ***Instrument effects***, there were no instrument effects in this experiment as the participants were given the same experimental task. As mentioned earlier in Chapter 6, two graphical tools were developed for the two groups to ensure that all participants worked within the same development environment.

7.6.2. External Validity

External validity was concerned with the extent to which the findings of our study could be generalised. ***Subjects***, these concerns were addressed by selecting a good sample size of engineers, as discussed in Chapter 6, Section 6.3.5. Although some experimenters argue that recruiting students might in itself be a threat to validity, many studies have showed that there is no difference between students and professionals (Sjøberg et al., 2005). ***Environment***, the environment effect was avoided by conducting the experiment at the same time in the same place. ***Task-related threats***, the experiment task was representative of industrial practice as it was an industrial control system and it used the common modelling language, 'SYSML',

in the design of the control systems. The duration of the task might also be considered a threat. However, we believe that the task duration was suitable as it was set based on the preliminary results of the experiment.

7.7. Discussion

A controlled experiment was conducted to evaluate the usability of the ICS-SES tool in terms of effectiveness, efficiency, and ease of task in helping developers to design secure control systems. The results obtained from the background questionnaire show that our tool meets a particular need as 81% of participants rate their cyber security knowledge as poor or very poor. The results revealed that participants lacked security training, which explains the result of their poor knowledge on ICS security issues, security standards and guidelines. Despite the existing security guidelines and best security design practise, which are published by several institutions and cyber security leaders worldwide, system engineers are not aware of these guidelines and methodologies. These findings are consistent with the results obtained from the qualitative study presented in Chapter 4.

Although participants have some awareness of the importance of considering security engineering throughout the system developing cycle, they are not aware of their role in building secure systems. However, the results show that participants are highly motivated to learn how to improve their security knowledge, particularly during their work. Moreover, the results from Google Analytics showed that the participants are highly engaged with the security training material, which is consistent with the results of participants' motivations towards improving their knowledge.

The performance results also show that participants will be better able to mitigate security vulnerabilities and reasoning regarding the appropriate secure design pattern with the help of the ICS-SES tool. Therefore, hypothesis H1 "Participants will be better able to solve a security problem in a system model with the help of ICS-SES tool" is accepted.

The result of knowledge assessment demonstrates that ICS-SES assists the improvement of the knowledge of participants, and makes it easier to perform the task of security engineering in comparison with the Plain group experience. Therefore, we can accept hypothesis H2 "Participants will better understand the

security problem with the support of the ICS-SES tool", hypothesis H3 "Participants will better understand the security solution with the help of the ICS-SES tool" and hypothesis H4 "the difficulty of solving the security problem will be less with the help of the ICS-SES tool".

In terms of efficiency, the result was significantly different ($P < 0.001$) as the participants take 52.16% less time to complete the task while using ICS-SES tool than the Plain group. Therefore, hypothesis H5 that "The time taken to solve the security problem given in the scenario will be less when using the ICS-SES tool" can be accepted.

The subjective feedback shows that the ICS-SES tool is generally useful and capable of being used within the work environment. Also, the participants were generally satisfied with the experience of the supported method. Engineer's subjective feedback regarding ICS-SES was positive, which is consistent with the results of the quantitative analysis.

The results obtained from the Cognitive Dimensions and subjective feedback revealed a list of suggestions to improve ICS-SES. Participants' recommendations were considered as potential improvements to the ICS-SES design.

7.8. Conclusion

This chapter introduced the evaluation of ICS-SES. A controlled experiment was executed in the Faculty of Technology at De Montfort University. 79 engineering students with different levels of education participated in the experiment. The ICS-SES tool was evaluated in terms of performance, effectiveness, efficiency and the ease of task. The data was collected, imported into the SPSS software suite and analysed.

The results of background survey clearly demonstrated that ICS developers lack security knowledge and training. However, the developers were motivated to learn and improve their knowledge through on-the-job training method. In addition, the results obtained from the comparison of the learning outcomes show the effectiveness of on-the-job-training support used in our tool and the results were statically significant.

The results obtained from the quantitative analysis of this study were presented in relation to the study hypotheses, as discussed in Section 7.7:

Hypothesis H_1 "Participants will be better able to solve a security problem in a system model with the help of ICS-SES tool" is accepted.

Hypothesis H_2 "Participants will better understand the security problem with the support of the ICS-SES tool" is accepted.

Hypothesis H_3 "Participants will better understand the security solution with the help of the ICS-SES tool" is accepted.

Hypothesis H_4"the difficulty of solving the security problem will be less with the help of the ICS-SES tool" is accepted.

Hypothesis H_5"The time taken to solve the security problem given in the scenario will be less when using the ICS-SES tool" is accepted.

The subjective feedback showed that the ICS-SES tool is generally useful and the users' feedback was positive.

Next chapter concludes the thesis and highlights the potential future directions of this research.

Chapter 8
Conclusion

This thesis presents a framework, Industrial Control System Security Engineering Support (ICS-SES), to assist developers in security engineering at the control system design phase. This chapter draws conclusions from both qualitative study that explored control system developers' needs for security engineering, and empirical study that examined the usability of the ICS-SES supported framework. The reminder of this chapter discusses the research contributions and outlines possible new directions for future work.

8.1. Conclusions

The thesis hypothesis is "Technology can be used to support developers in designing secure control systems and improve their security knowledge." The resulting supported framework was defined as the Industrial Control Systems Security Engineering Support, 'ICS-SES'. The argument is that ICS-SES can assist engineers to develop secure control systems. ICS-SES is usable and can effectively help developers to improve their security knowledge and design secure control systems.

Based on academic and industrial motivation, discussed in Chapter 1.2, and the findings from the qualitative study on ICS developers' needs regarding security engineering, as discussed in Chapter 4, the ICS-SES framework was proposed in Chapter 5, with the aim of supporting the development of control system security by design. In particular, it guides users in mitigating any detected security flaws in a system model and provides adaptive training material tailored to users' needs. The framework was evaluated through empirical study, as presented in Chapter 6, by assessing its usability in assisting engineers to develop secure control systems in terms of effectiveness, efficiency and ease of task. An empirical study with a group of seventy-nine engineers with different educational levels found that the ICS-SES tool can improve security awareness and knowledge and help to solve design security problems with fewer difficulties in a reduced amount of time, as discussed in Chapter 7.

8.1.1. Research question 1

The answer to research question 1 *"What is the state-of-the-art in control system security engineering?"* was obtained by conducting qualitative studies. In Chapter 2, the related literature was systematically reviewed. This study revealed that there is a knowledge gap in the control system security engineering research area that requires greater attention from researchers. The literature ascertained that including security over the entire development life cycle is paramount to building secure control systems that are resistant to attack. The study findings clearly demonstrated that control system development processes lack security considerations. It also showed that control system developers lack security awareness and knowledge, and thus that there is a culture gap between system developers and security experts. This gap was better understood and explained through the results of the research interviews conducted with control system developers, as examined in Chapter 4. The results of these interviews showed that developers lack technical and training support in terms of security. Lack of knowledge and support were also confirmed from the results of the empirical study presented in Chapter 7. As a result, control system security engineering is still not sufficient and more effort is needed from researchers to bridge the gap between control system developers and security experts.

8.1.2. Research question 2

Research question 2 *"What are developers' needs regarding the design of secure control systems?"* was answered by the results obtained from the exploratory research interviews studied in Chapter 4. A qualitative study was conducted to assess developers' support needs for the design of secure systems. In line with the results of the systematic literature review, the study findings revealed two key reasons for the lack of security consideration throughout the system development cycle: lack of knowledge and lack of support. In addition, the results gave further insights into developers' requirements in terms of developing system security by design. Control system developers do not consider security requirements throughout the system design phase; the focus is purely on functionality and safety. However, developers have some awareness of the necessity of security and showed a readiness to improve their knowledge, which is consistent with the participants' motivations shown in the

experimental study presented in Chapter 7. The study results revealed that control system engineers need both technical support that can assist in developing control system security by design, and training support that provides fundamental required security knowledge.

8.1.3. Research question 3

The answer to research question 3 *"Can an on-the-job adaptive training tool be created to support control system security by design?"* is yes. In Chapter 5, an adaptive supported framework was created and named 'Industrial Control System Security Engineering Support (ICS-SES)', which included a pattern-based security guide and tailored security training. A prototyping tool for ICS-SES was created with an emphasis on these two main functionalities in order to evaluate the feasibility of our supported method. After the preliminary evaluation presented in Chapter 6, Chapter 7 evaluated the improved ICS-SES tool by conducting a controlled experiment with a large user group, consisting of seventy-nine engineers, to find out whether ICS-SES can help developers to improve their knowledge and design secure control systems. The results showed that almost all participants were pleased with the tool's support and expressed their feelings that the tool helped to solve and understand security problems.

8.1.4. Research question 4

Research question 4, *"Can a supported tool assist developers in designing secure control systems?"* was successfully answered through the work performed in Chapter 7. The usability of the ICS-SES tool was evaluated by conducting an empirical study, presented in Chapter 6, using seventy-nine engineers with a variety of educational levels. Participants were divided into two groups: the experimental group, which used the ICS-SES tool, and the control group, which did not. The two groups were compared in terms of participants' abilities to understand the security problem context (the security flaw in the system model provided in the experiment task), reasoning regarding security patterns to solve the problem, efficiency in identifying an appropriate security pattern to mitigate the security weakness, and the ease of the task of finding a suitable solution to the problem. We were able to obtain statistically significant differences in terms of participants' performances and learning. The

group using the ICS-SES tool performed better than the control group. We were able to obtain statically significant efficiency (time) differences between the two groups. The group with ICS-SES spent less time performing the task than the control group. We were also able to obtain statically significant differences as to the ease of performing the task, and noted that the group with the supported tool found it easier. In addition, the subjective feedback obtained showed that participants were pleased with the supported tool and their overall commentary regarding ICS-SES was positive.

8.2. Contributions

A novel method to support designing secure industrial control systems. The ICS-SES method brings together patterns from security expertise and serves them to system developers in an interactive manner. It also contributes to education process in terms of security, as it provides a systematic way to educate system developers in secure design patterns. This has implications for improving the security of industrial control system design.

An on-the-job practical guide for security patterns in combination with training aid. A security patterns guide has been proposed by a number of researchers (Weiss and Mouratidis, 2008) (Hasheminejad and Jalili, 2009) (Fernandez et al., 2011) using different pattern classifications. However, this is the first method that provides a practical and systematic guide on reasoning about security patterns through an on-the-job approach combined with tailored security training material.

On-the-job security training in the control engineering discipline. The on-the-job training approach is commonly used in the engineering field as engineers typically learn through practice, by 'doing' or observing at the work site (Rooney et al., 2014). ICS-SES contributes to control engineering education by employing this approach to educate engineers in security engineering. There is further contribution from the use of a problem-based learning strategy, which has been applied in engineering education for many years in a variety of professional engineering schools (Jonassen and Hung, 2008, Mills and Treagust, 2003), in the security training provided by ICS-SES.

Contextualised and personalised security training for industrial control systems. ICS-SES provides security training material that is adaptive to the system design security problem context and tailored to users' training needs by employing an automated planner technique. Traditionally, automated planners have been used in e-learning course design in combination with Case-Based Reasoning (CBR) to save predefined learning plans in a library, where learning plans are recurrent (Garrido et al., 2012). However, our method uses an automated planner to interactively generate learning plans that are different from one trainee to another within a given work environment; we have not been able to identify similar use anywhere in the area of security.

The potential for improving security awareness and knowledge of control system engineers. Through an experimental study conducted in this research, it was found that ICS-SES helps engineers to improve their security skills and transfer lessons learned through ICS-SES to similar problems. This finding shows that the ICS-SES contributes to security education.

Improving comprehension of control system security. Through an experimental study conducted in this research, it was found that ICS-SES can help to reduce difficulties in understanding control system vulnerabilities and related security solutions, and further that our results are statically significant. ICS-SES contributes to helping tackle the difficulties faced by control engineers in the field of security engineering. It also helps improve the usability of security standards and guidelines in practice.

8.3. Directions for Future Work

Whilst this thesis has demonstrated the potential for effectively improving industrial control system security engineering by providing an educational supported method, there are still opportunities for extending the scope of this research. This section discusses a number of such possible directions.

As a part of the ICS-SES process, the tool uses users' feedback, which is collected after pattern application and assessment in the Case-Based Security Patterns (CBSP), so as to enhance the pattern selection process. Future work can address the impact of changes on other concerns such as performance, safety, cost, etc. ICS-SES can be

extended to consider impact restrictions in the selection algorithm used in order to be compatible with aspects of system functionality.

The ICS-SES tool was based on an automated planning technique in order to provide security training support. While the current training aid has achieved significant positive results during the tool evaluation phase, the tool could be further extended to apply an approach for monitoring the execution of training plans that could help to improve their validity. Trainees can be monitored to avoid any incidents that might occur during training regarding their performance, such as many trainees being unable to achieve topic competence, or efficiency, such as trainees taking a longer time than expected to complete a particular training topic.

Currently, the ICS-SES tool supports control system developers at the design phase. Future work could focus on supporting security engineering during other development phases. For example, in software engineering, there is an educational tool named ESIDE, as proposed in reference (Pilkington et al., 2009), that can help software engineers to write secure code. Future work could adapt this tool to be compatible with the nature of control systems to support the implementation of secure control systems.

8.4. Closing remarks

This work reveals that it is possible to develop industrial control system security by design by supporting system developers and improving their security knowledge. In this thesis, the Industrial Control System Security Engineering Support (ICS-SES) framework was proposed, with the aim of assisting developers in the design of control systems through pattern-based security guides and personalised security training support. The empirical evaluation study shows that ICS-SES can help to develop control system security by design and improve developers' security awareness and knowledge. The experience gained from this PhD project provides sufficient foundation work to generalise the use of this supported method in the industrial control system domain.

Appendices

Appendix A : Systematic Literature Review (Appendix to Chapter 2)

TABLE-A-1: PUBLICATION COLLECTED IN SYSTEMATIC LITERATURE REVIEW

ID	Publisher	Year	Title
L1	IEEE	2011	A PROPOSED ARCHITECTURE FOR SCADA NETWORK SECURITY
L2	IEEE	2013	A Proposed Australian Industrial Control System Security Curriculum
L3	IEEE	2009	Advanced Key-Management Architecture for Secure SCADA Communications
L4	IEEE	2014	Using Integrated System Theory Approach to Assess Security for SCADA Systems Cyber Security for Critical Infrastructures: A Pilot Study
L5	IEEE	2010	ChallengesandIssuesin the SystemArchitectingfor SystemicRisk Infrastructure System:AnIndustrialCase Study
L6	IEEE	2012	Challenges and opportunities in securing industrial control systems
L7	IEEE	2015	Analysis of Cyber Security for Industrial Control Systems
L8	IEEE	2013	Current issues and challenges on cyber security for industrial automation and control systems
L9	IEEE	2014	Cyber Security Issues of Critical Components for Industrial Control System
L10	IEEE	2016	Cyber Security of Cyber Physical Systems: Cyber Threats and Defense of Critical Infrastructures
L11	IEEE	2013	Design and Development of Wireless RTU and Cybersecurity Framework for SCADA System
L12	IEEE	2010	Designing Secure SCADA Systems Using Security Patterns
L13	IEEE	2013	Developing a Critical Infrastructure and Control Systems Cybersecurity Curriculum
L14	IEEE	2010	Applying Lessons from Safety-Critical Systems to Security-Critical Software
L15	IEEE	2008	Functional Safety and System Security in Automation Systems – A Life Cycle Model
L16	IEEE	2016	Improving Cybersecurity for Industrial Control Systems
L17	IEEE	2013	Industrial Control Systems Security: What is happening?
L18	IEEE	2008	Information Security Challenges in Industrial Automation Systems
L19	IEEE	2014	Insights on the Security and Dependability of Industrial Control Systems
L20	IEEE	2011	AVATAR: A SysML Environment for the Formal Verification of Safety and Security Properties
L21	IEEE	2016	Model-Driven Engineering for Designing Safe and Secure Embedded Systems
L22	IEEE	2015	Overview of Cyber-security of Industrial Control System
L23	IEEE	2013	RECENT DEVELOPMENTS IN STANDARDS AND INDUSTRY SOLUTIONS FOR CYBER SECURITY AND SECURE REMOTE ACCESS TO ELECTRICAL SUBSTATIONS
L24	IEEE	2015	Replacing Fear with Knowledge - Cyber Security for Substation Automation, Protection and Control Systems
L25	IEEE	2013	ReviewofSecurityIssuesinIndustrialNetworks
L26	IEEE	2015	Secure Design Patterns for Security in Smart Metering Systems
L27	IEEE	2008	Security for process control systems
L28	IEEE	2014	Software Security Assurance of Electrical Grid Systems
L29	IEEE	2015	SysML-Sec A Model Driven Approach for Designing Safe and Secure Systems
L30	IEEE	2016	The Cybersecurity Landscape in Industrial Control Systems
L31	IEEE	2016	The Dilemma of Securing Industrial Control Systems UAE Context
L32	IEEE	2016	The Security Challenges in the IoT enabled Cyber-Physical Systems and Opportunities for Evolutionary Computing & Other Computational Intelligence
L33	IEEE	2014	Using Cybersecurity as an Engineering Education Approach on Computer Engineering to Learn About Smart Grid Technologies and the Next Generation of Electric Power Systems
L34	ACM	2013	A Systems Approach to Cyber Assurance Education
L35	ACM	2016	Addressing Critical Industrial Control System Cyber Security Concerns via High Fidelity Simulation
L36	ACM	2011	Critical Infrastructure Security Curriculum Modules
L37	ACM	2015	Guiding the selection of security patterns based on security requirements and pattern classification
L38	ACM	2015	Industry Cybersecurity Workforce Development
L39	ACM	2009	On Building Secure SCADA Systems using Security Patterns
L40	ACM	2015	Security and Privacy Challenges in Industrial Internet of Things
L41	ACM	2013	Security-Aware, Model-Based Systems Engineering with SysML
L42	ACM	2014	Software Engineering Issues Regarding Securing ICS: An Industrial Case Study
L43	ELSEVIER	2013	A DEVELOPMENT FRAMEWORK FOR SOFTWARE SECURITY IN NUCLEAR SAFETY SYSTEMS: INTEGRATING SECURE DEVELOPMENT AND SYSTEM SECURITY ACTIVITIES
L44	ELSEVIER	2015	A survey of cyber security management in industrial 2015

L45	ELSEVIER	2016	Model-based security engineering for cyber-physical systems: A systematic mapping study
L46	ELSEVIER	2010	SCADA System Cyber Security – A Comparison of Standards
L47	ELSEVIER	2016	
L48	Springer	2015	Teaching Industrial Control System Security Using Collaborative Projects
L49	eBook	2014	Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems
L50	eBook	2012	Railway Safety, Reliability, and Security: Technologies and Systems Engineering: Technologies and Systems Engineering
L51	*ENISA*	2014	Certification of Cyber Security skills of ICS/SCADA professionals Good practices and recommendations for developing harmonised certification schemes
L52	*ENISA*	2015	Cyber (In-)security of Industrial Control Systems: A Societal Challenge
L53	GCCS	2015	Cyber Security of Industrial Control Systems
L54	CPNI	2011	CYBER SECURITY ASSESSMENTS OF INDUSTRIAL CONTROL SYSTEMS A GOOD PRACTICE GUIDE
L55	*ENISA*	2015	Securing Industrial Control Systems Secure. Vigilant. Resilient
L56	ENISA	2015	Security Awareness Compliance Requirements
L57	*ENISA*	2011	Protecting Industrial Control Systems Annex I: Desktop Research Results [Deliverable – 2011-12-09]
L58	Homeland Security	2011	Common Cybersecurity Vulnerabilities in Industrial Control Systems
L59	ENISA	2015	Analysis of ICS-SCADA Cyber Security Maturity Levels in Critical Sectors
L60	CPNI	2015	SECURITY FOR INDUSTRIAL CONTROL SYSTEMS IMPROVE AWARENESS AND SKILLS A GOOD PRACTICE GUIDE
L61	Homeland security	2016	Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies
L62	NIST	2015	NIST Guide to Industrial Control Systems (ICS) Security
L63	ENISA	2011	Protecting Industrial Control Systems
L64	SANS	2015	Secure Architecture for Industrial Control Systems
L65	SANS	2015	The State of Security in Control Systems Today

Appendix B : Ethical Approval

Appendix B-1: Ethical approval for Interviews

From: **Anne Smith** <AmSmith@dmu.ac.uk>
Date: 22 November 2013 09:59
Subject: Ethics Application Nuria BENJUMA 1213/185
To: Research Students <researchstudents@dmu.ac.uk>, Helge Janicke <heljanic@dmu.ac.uk>
Cc: "P12188805@myemail.dmu.ac.uk" <P12188805@myemail.dmu.ac.uk>

Dear All

Please note that Nuria's ethical application has been approved.

Kind regards

Anne

Anne Smith
Research Co-ordinator
Research & Innovation Office (4.64)
Faculty of Technology

DE MONTFORT UNIVERSITY
Gateway Building
The Gateway
Leicester LE1 9BH
UK
T: +44 (0) 116 250 6519
E: amsmith@dmu.ac.uk
W: dmu.ac.uk

Appendix B-2: Ethical approval Experiment

From: Anne Smith <AmSmith@dmu.ac.uk>
Sent: Thursday, November 24, 2016 11:51 AM
To: Nuria Benjuma
Cc: Helge Janicke; Richard Smith; Research Students
Subject: Ethics Application - Nuria BENJUMA 1415/247-1

Dear Nuria

Research Ethics Application Ratification Required: 1415/247-1 *Building secure industrial control systems using security patterns*

Your application to gain ethical approval for research degree activities has been considered and APPROVED by the Faculty Research Ethics Committee (FREC) on 14 November 2016. No further issues were raised by the committee.

Please be aware that changes to the project plan or unforeseen circumstances may raise ethical issues. If this is the case it is the researcher's duty to repeat the ethics approval process.

Kind regards

Anne

Anne Smith
Research Coordinator
Research & Innovation Office (GH 4.64)
Faculty of Technology

DE MONTFORT UNIVERSITY
Gateway Building
The Gateway
Leicester LE1 9BH
UK
T: +44 (0)116 250 6519
E: amsmith@dmu.ac.uk
W: dmu.ac.uk

Appendix C: Research Interviews (Appendix to Chapter 4)

C 1. Interview Questions

Introduction

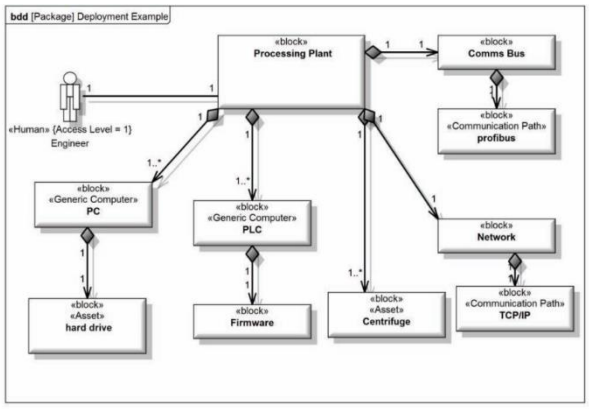
The main purpose of this interview is to collect qualitative data that would explore the current security knowledge of control system developers and identify their security training needs. This interview structured as it is shown in the following table.

<table>
<tr><td colspan="2">Section 1: Security awareness and knowledge</td></tr>
<tr><td colspan="2">Q1- What are security issues related to Industrial Control Systems field?</td></tr>
<tr><td colspan="2">Q2- In your opinion, at which phase of development cycle security concerns should be involved?</td></tr>
<tr><td colspan="2">Q3- How do you determine whether your system design is secure?</td></tr>
<tr><td colspan="2">Q4- From your perspective, what are the most important security rules that developers should follow in order to design secure systems?</td></tr>
<tr><td rowspan="2">Q5- Example discussion

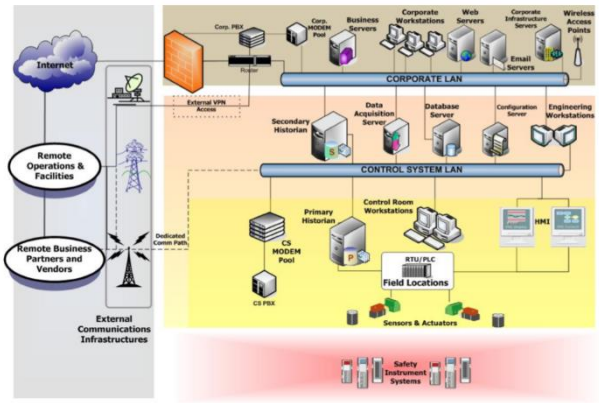
If you design this system where will you consider security policies?</td><td>If you design your system using SysML, use <u>Example 1</u></td></tr>
<tr><td>If you do not use SysML, use <u>Example 2</u></td></tr>
<tr><td colspan="2">Q6- Do you know secure design patterns or guidelines?</td></tr>
<tr><td colspan="2">Section 2: Available support for ICS security engineering</td></tr>
<tr><td colspan="2">Q7- How do you select the security patterns?</td></tr>
<tr><td colspan="2">Q8-What do you use for modelling? Does that tool support system security?</td></tr>
</table>

Q9- Have you attended any security training program?	YES	9.1 Did that training help you to improve your security awareness and design secure systems?
	NO	9.2 Do you think you need security training? Why?
Section 3: Developers' needs and requirements for designing secure systems		
Q10- What kind of support could improve security knowledge of control engineers?		
Q11- What are features that would make a training tool more useful for engineers?		
Q12- What are features of training tools that distract from learning?		

System Example-1: for SysML/UML users:



System Example-2: For users who do not use SysML/UML.



C 2. Interview Analysis - the preliminary results

No support
Tool doesn’t support applying security.
Sysml doesn’t include security information.
I cannot model security.
Tool doesn’t support security.
Security should be at the beginning of the design stage
the problem none of the tools allow you to do that
The focus is on control not on security
in the past security wasn’t an issue
The connectivity
Connectivity to the internet
Connectivity carries risk
engineers and operators.
Protocols are not secure
Industrial internet of things (industry 4.0)
multi layers of security need to be investigated
The key is connectivity
Systems that controlled from central point of view
Security is complex
The connectivity leads to security concerns
The access should be limited
The main issue is how to stay up to date
The gap between engineers and security experts
structure and operation security issues
The key is system structure
using ready protocols
unable to customise or modify protocols
dealing with safety only
difficult to say what is vulnerable and where can consider security
systems should be secure
the background is far away from security
haven’t used security patterns
Engineers don’t have knowledge in security
no check against security
System is tested just for functionality and performance
the rely is on other security like PC protection and firewalls
no security measurements
Engineers should have a good knowledge on security
Not aware of security patterns /standards
Security should be implemented at the beginning.
Security can be added at the end
Engineers based on vendors for security
Measurement is for safety only
Engineers don’t if system is secure.

Engineers don’t have security background
Don’t need training support.
Engineers care about functionality more than security
Support should be during work
Engineers should understand security basics and principles
available training is very general
Workshops with industry are general
No security training
No security in engineering education. p4
available training is not specific for design. P5
have never been thought how to protect control systems
Every engineer should have security awareness.
security knowledge should be improved
Security training is important
training can make everybody implement security
Engineers should have a good knowledge on security
Regular courses are good for security training
having more control of security
need tool that gives a capability to implement security
Provide examples
Explain technical details
provide challenges and exercise
trying not only reading
avoid too much text
know the consequences or risks of vulnerability
system designers should be connected with security experts
Engineers should work together with security experts
Explain and describe the problem and risks
Avoid anonymous information
Avoid unclear messages
System designers is the key of control system security
Avoid security jargon
Use standards
Training must be based on prior knowledge
Provide suggestion to choose from
tools only give feedback about problem
explain how to fix problem
engineers need security training
Engineers need automated support
Save our time
good simulation
Drag and drop
A good GUI.
Good layout
Provide help and assistance
Provide high level and conceptual explanation
Simple and effective
Avoid over complicated

provide personal support
Don’t bother
Regular training
educate engineers in system weaknesses
Provide support to improve security
Tool should be easy to use
Provide clear steps and explanation help
suggested some solutions. P6
User friendly interface
Provide a template of solution
Tool that can be used and configured
provide solutions according to the problem
Give some description on how solutions work
Provide learning material
Use domain specific language
Show diagrams and examples

Appendix D : The empirical experiment (Appendix to Chapter 6)

D.1 Pre-Questionnaire

Pre-Questionnaire

This questionnaire is an essential part of PhD research conducted by Nuria Benjuma at De Montfort University, UK. The research aims to develop a security training method to support control systems' developers during system design. It also aims to employ the technology in providing on-the-job security training based on the personal needs.

Thank you for your time. Your answers are very valuable and important to complete this research successfully.

*Required

Section-A: Security Background

1. **Your Reference Number** *

2. **How do you rate your knowledge about common security problems in industrial control systems?** *
Mark only one oval.

	1	2	3	4	5	
Very poor	◯	◯	◯	◯	◯	Excellent

3. **How do you rate your knowledge about security standards and guidelines for industrial control system design?** *
Mark only one oval.

	1	2	3	4	5	
Very poor	◯	◯	◯	◯	◯	Excellent

4. **When you design a control system, do you take security requirements into consideration?** *
Mark only one oval.

	1	2	3	4	
Never	◯	◯	◯	◯	Always

5. **Are you responsible for the security in the control system development?** *
Mark only one oval.

- ◯ Yes
- ◯ Yes, but shared with others
- ◯ No
- ◯ Other: _____

6. **At which phase of control system development cycle security should be considered?** *
Mark only one oval.

- ◯ Design (concept and design)
- ◯ Build (Testing – Installation – Site acceptance)
- ◯ Operation (Operation – Maintenance – Modification)
- ◯ All the above phases
- ◯ I don't know

7. **For each one of the following security guidelines, please indicate :** *
Mark only one oval per row.

	You never heard of it	You have heard of it but haven't used it	You used it
NIST	◯	◯	◯
CPNI	◯	◯	◯
DHS	◯	◯	◯
SANS	◯	◯	◯
ICS-CERT	◯	◯	◯
NISA	◯	◯	◯
ISA	◯	◯	◯
NCCIC	◯	◯	◯

8. **Have you had any training on control system security before?** *
Mark only one oval.

- ◯ Yes *Skip to question 9.*
- ◯ No *Skip to question 11.*
- ◯ Don't want to say *Skip to question 11.*

Section-B: Security Training

9. **When did the training take place training?** *
Mark only one oval.

- ◯ 1-5 years ago
- ◯ Last month
- ◯ Last week
- ◯ Other: _____

10. **Where did the training take place?** *
Mark only one oval.

- ◯ At work
- ◯ During study
- ◯ Other: _____

Section-C : User Motivation

11. **If you develop a system and discover that the system has a security weakness, which of the following describes your most common action?** *
Mark only one oval.

- ◯ Learn what is the problem and how to fix it
- ◯ Ignore the problem and continue working
- ◯ Pass it to different team in the organisation
- ◯ Other: _____

12. **In terms of learning new skills, which one of the following training methods do you prefer?** *
Mark only one oval.

- ◯ Regular training courses
- ◯ Learn while you are working (On-The-Job-Training)
- ◯ Workshops and discussion
- ◯ Other: _____

Section-D: Pre-Test

If you design an industrial control system and using a security analyser to discover any security weakness or vulnerable component in your system model. If the security analyser identifies a security problem which is as following:

The technicians can use HMI (Human Machine Interaction) to set and stop Alarms. But this also allows them to send commands to PLCs, while they should not do.

Please answer the following questions with regards to the above scenario:

13. **The problem identified in the above scenario is called:** *
Mark only one oval.

- ◯ Low factor authentication
- ◯ Improper Input validation
- ◯ Poor Code quality
- ◯ Improper Authentication
- ◯ Planning/Policy/Procedures issue
- ◯ Improper access control
- ◯ Weak Firewall Rules
- ◯ Insufficient Verification of Data Authenticity
- ◯ Cryptographic Issue
- ◯ I do not know

14. **Which one of the following security patterns can be integrated to the system to reduce the problem?** *
Mark only one oval.

- ◯ Virtual Private Network (VPN)
- ◯ Secure Logger
- ◯ Firewall
- ◯ Critical Functions with Network Security Zones and Layers
- ◯ Attribute-Based Access Control
- ◯ Change All Default Passwords and Require Strong Passwords
- ◯ Security Incident Handling Process
- ◯ Attempts to log on with invalid passwords are limited
- ◯ Role- Based Access Control (RBAC)
- ◯ Security Management Program
- ◯ Sensitive Information Management
- ◯ I do not know

D.2 Post-Questionnaire (A)

Post-Questionnaire- Group A

Note: Please answer the following questions with regards to your experiences on the ICS-SES tool that you used in the previous part of the evaluation survey.

*Required

Section A: Participants' Experience

1. **Your Reference Number** *

2. **With regards to solving the given security issue, please rate your agreement with these statements:** *
Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I have no difficulty in understanding the security problem	◯	◯	◯	◯	◯
I have no difficulty in finding a possible solution	◯	◯	◯	◯	◯
I have no difficulty in understanding the solution	◯	◯	◯	◯	◯

3. **Did you solve the security problem given in the scenario?** *
Mark only one oval.

- ◯ Yes
- ◯ No
- ◯ I am not sure

4. **(if No) Why you didn't solve the problem? please indicate why do you think this happens:**

5. **Did you use the ICS-SES tool for solving the problem?** *
Mark only one oval.

- ◯ Yes *After the last question in this section, skip to question 7.*
- ◯ No *After the last question in this section, skip to question 12.*
- ◯ Not sure *After the last question in this section, skip to question 12.*

6. **(if No) Why you did not use the ICS-SES tool?**
Mark only one oval.

- ◯ I know how to solve the problem.
- ◯ Other: _____

Section B: Usability Evaluation Framework Cognitive Dimensions(CD) of ICS-SES Tool

7. **With regards to the Usability of the ICS-SES tool, please rate your agreement with these statements:** *
Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The tool allows me to access all of the relevant information easily	◯	◯	◯	◯	◯
The tool aided in solving hard or complex problems that would not have been possible in my head	◯	◯	◯	◯	◯
The training material provided the full range of information required to solve the problem	◯	◯	◯	◯	◯
The tool accurate portrays the situation in a context engineers are familiar with	◯	◯	◯	◯	◯
The information provided is consistent across topics	◯	◯	◯	◯	◯
The tool allows me to understand why security vulnerabilities occur within engineering designs	◯	◯	◯	◯	◯

Section C: Usefulness and Satisfaction

8. **With regards to the usefulness of the ICS-SES tool, please rate your agreement with these statements:** *
Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Overall, it is useful	◯	◯	◯	◯	◯
It helps me to understand the problem	◯	◯	◯	◯	◯
It helps me to understand how to solve the problem	◯	◯	◯	◯	◯
The tool is easy to use	◯	◯	◯	◯	◯
The training materials meet my personal needs to understand related security topics	◯	◯	◯	◯	◯
The training material is easy to understand	◯	◯	◯	◯	◯
The tool can help developers in designing secure control systems	◯	◯	◯	◯	◯

9. **With regards to the satisfaction with the training tool, please rate your agreement with these statements:** *

Mark only one oval per row.

	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree
I am satisfied with it	◯	◯	◯	◯	◯
I need to use it to design more secure systems	◯	◯	◯	◯	◯
I am satisfied with the support training material	◯	◯	◯	◯	◯
I can use it easily during every day work without distraction	◯	◯	◯	◯	◯

10. **What do you like best about the ICS-SES tool?**

11. **What do you like least about the ICS-SES tool?**

Section D: Post-Test

If you design an industrial control system and using a security analyser to discover any security weakness or vulnerable component in your system model. If the security analyser identifies a security problem which is as following:

The technicians can use HMI (Human Machine Interaction) to set and stop Alarms. But this also allows them to send commands to PLCs, while they should not do.

Please answer the following questions with regards to the above scenario:

12. **The vulnerability identified in the above scenario is:** *

Mark only one oval.

- ◯ Low factor authentication
- ◯ Improper input validation
- ◯ Poor Code Quality
- ◯ Improper Authentication
- ◯ Planning/Policy/Procedures issue
- ◯ Improper access control
- ◯ Weak Firewall Rules
- ◯ Insufficient Verification of Data Authenticity
- ◯ Cryptographic Issue
- ◯ I do not know

13. **Which of the following security patterns can be integrated to the system to reduce the problem?** *

Mark only one oval.

- ◯ Virtual Private Network (VPN)
- ◯ Secure Logger
- ◯ Firewall
- ◯ Critical Functions with Network Security Zones and Layers
- ◯ Attribute-Based Access Control
- ◯ Change All Default Passwords and Require Strong Passwords
- ◯ Security Incident Handling Process
- ◯ Attempts to log on with invalid passwords are limited
- ◯ Role- Based Access Control (RBAC)
- ◯ Security Management Program
- ◯ Sensitive Information Management
- ◯ I do not know

D.3 Post-Questionnaire (B)

Post-Questionnaire-Group B

Note: Please answer the following questions with regards to your experiences on solving the security problem given in the previous part of the evaluation survey.

*Required

Section A: Participants' Experience

1. **Your Reference Number** *

2. **With regards to solving the security issue given in the scenario, please rate your agreement with these statements:** *
Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I have no difficulty in understanding the security problem	◯	◯	◯	◯	◯
I have no difficulty in finding a possible solution	◯	◯	◯	◯	◯
I have no difficulty in understanding the solution	◯	◯	◯	◯	◯

3. **With regards to finding related information, please rate your agreement with these statements:**
Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I easily found related information about the problem	◯	◯	◯	◯	◯
I easily found related information about how to solve the problem	◯	◯	◯	◯	◯
I found required information to meet my personal needs to understand related security topics	◯	◯	◯	◯	◯

4. **Did you solve the security problem given in the scenario?** *
Mark only one oval.

◯ Yes
◯ No *After the last question in this section, skip to question 6.*
◯ I'm not sure *After the last question in this section, skip to question 6.*

5. **What did help you to solve the security problem?** *
Mark only one oval.

- ◯ Your security background *Skip to question 7.*
- ◯ Online resources *Skip to question 7.*
- ◯ Other: _____ *Skip to question 7.*

Section B: Difficulties

6. **What was the most difficult task in solving the security problem?** *
Mark only one oval.

- ◯ Understanding the problem
- ◯ Finding a solution
- ◯ Understanding the solution
- ◯ Finding the right keywords to search
- ◯ Huge information on online resources
- ◯ All of the above
- ◯ Other: _____

Section C: Support Needs

7. **With regards to the security issue, please rate your agreement with these statements:** *
Mark only one oval per row.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I need to understand the problem	◯	◯	◯	◯	◯
I need to improve my security skills	◯	◯	◯	◯	◯
I need guidance to solve the security problem in the system model	◯	◯	◯	◯	◯
I need training material that meets my needs in the problem context	◯	◯	◯	◯	◯
I need more support in designing secure system	◯	◯	◯	◯	◯

8. **Can you think of obvious ways that could help you to improve your security knowledge? What are they?**

_____

_____

_____

_____

_____

9. **Please write any additional comments:**

Section D: Post-Test

If you design an industrial control system and using a security analyser to discover any security weakness or vulnerable component in your system model. If the security analyser identifies a securi problem which is as following:

The technicians can use HMI (Human Machine Interaction) to set and stop Alarms. But this also allows them to send commands to PLCs, while they should not do.

Please answer the following questions with regards to the above scenario:

10. **The vulnerability identified in the above scenario is:** *

Mark only one oval.

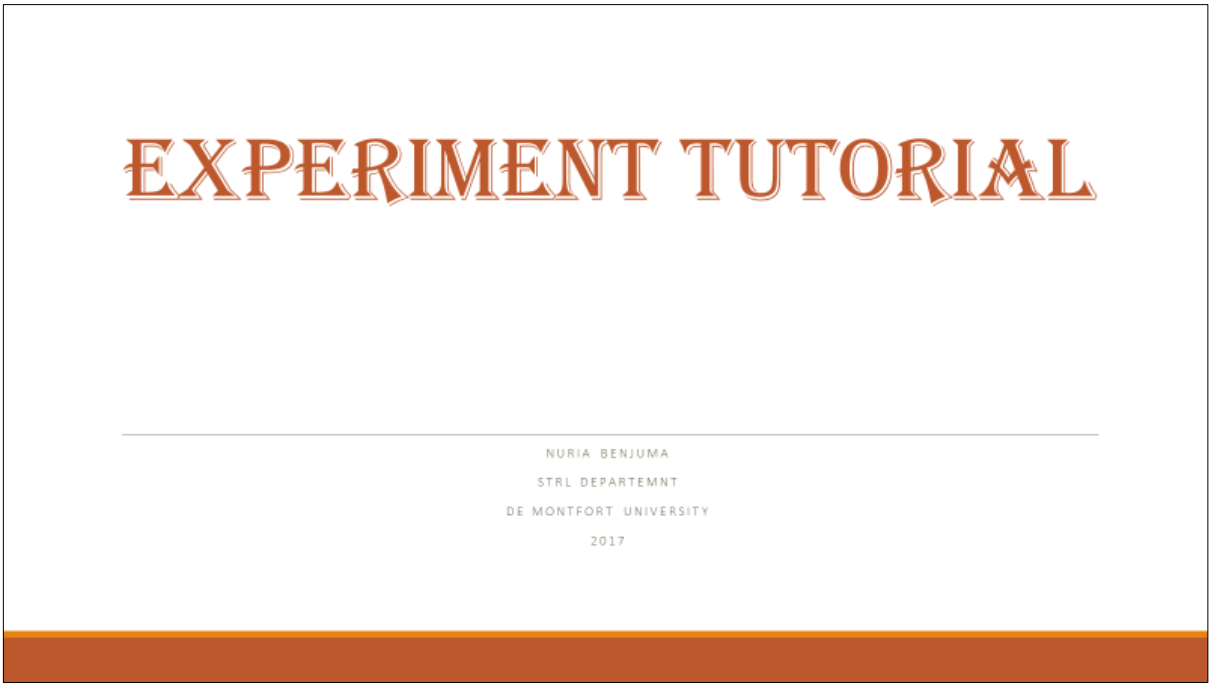
- ◯ Low factor authentication
- ◯ Improper input validation
- ◯ Poor Code Quality
- ◯ Improper Authentication
- ◯ Planning/Policy/Procedures issue
- ◯ Improper access control
- ◯ Weak Firewall Rules
- ◯ Insufficient Verification of Data Authenticity
- ◯ Cryptographic Issue
- ◯ I do not know

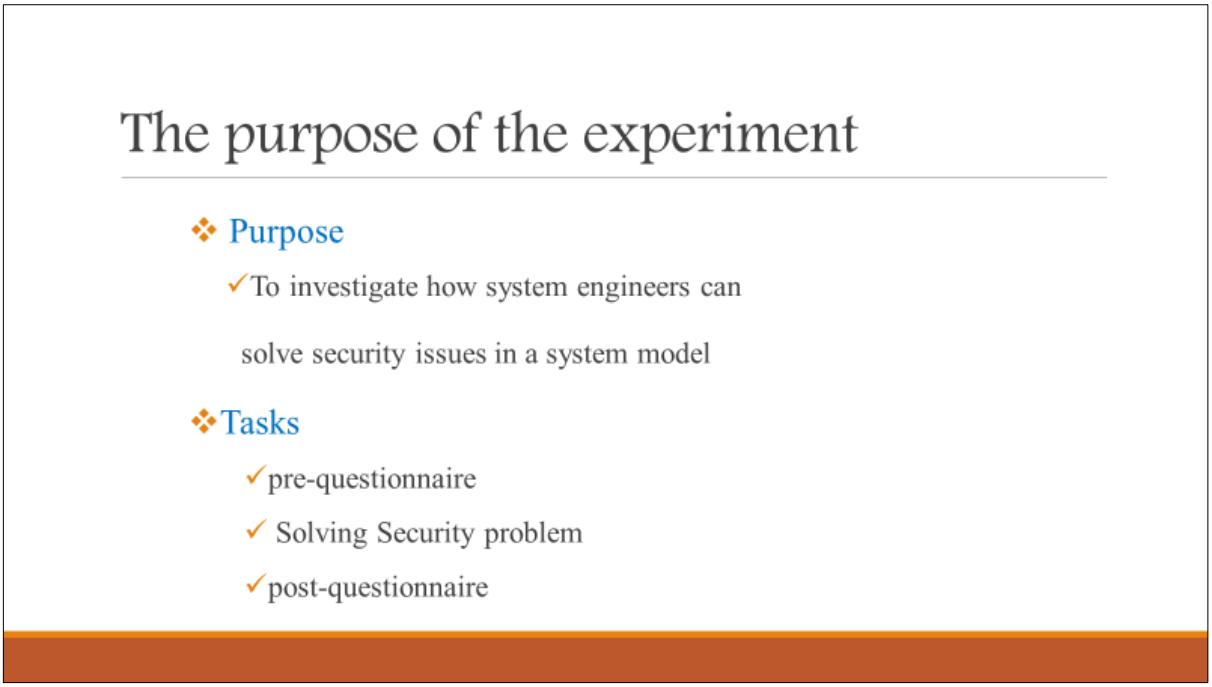
11. **Which of the following security patterns can be integrated to the system to reduce the problem?** *

Mark only one oval.

- ◯ Virtual Private Network (VPN)
- ◯ Secure Logger
- ◯ Firewall
- ◯ Critical Functions with Network Security Zones and Layers
- ◯ Attribute-Based Access Control
- ◯ Change All Default Passwords and Require Strong Passwords
- ◯ Security Incident Handling Process
- ◯ Attempts to log on with invalid passwords are limited
- ◯ Role- Based Access Control (RBAC)
- ◯ Security Management Program
- ◯ Sensitive Information Management
- ◯ I do not know

D.4 Tutorial

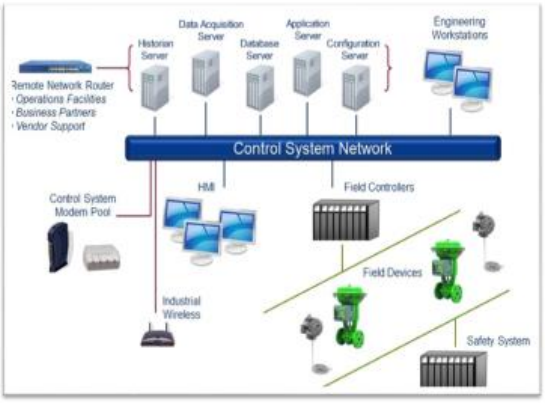

EXPERIMENT TUTORIAL
NURIA BENJUMA
STRL DEPARTEMNT
DE MONTFORT UNIVERSITY
2017


The purpose of the experiment
Purpose
To investigate how system engineers can
solve security issues in a system model
Tasks
pre-questionnaire
Solving Security problem
post-questionnaire

Industrial Control System (An overview)

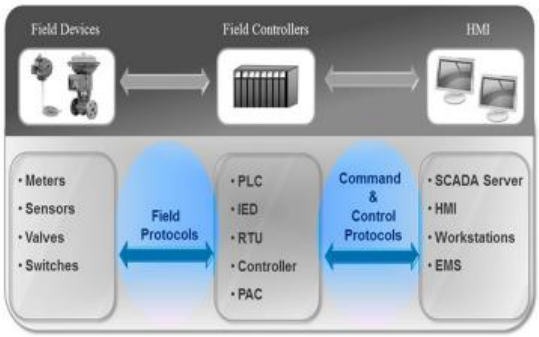
Typically used for remotely monitoring and controlling critical infrastructures such as

- Water and wastewater treatment
- Chemicals
- Oil and natural gas
- Transportation
- Power station



Common Industrial control Systems components

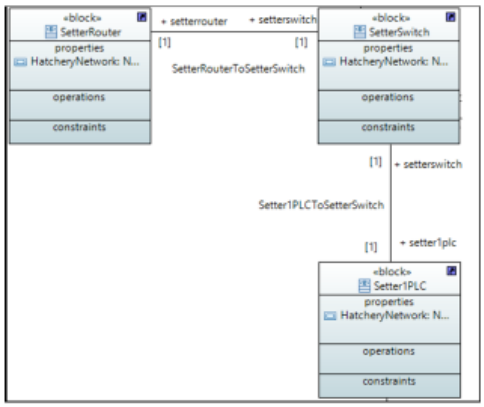
- ***Field Devices:*** are the interface between physical processes and control system.
- ***Field Controllers:*** run the communication between field devices and Human Interaction Interface (HMI).
  - Collect input and output data from field devices
  - Send data to HMI
- ***Human Machine Interface (HMI):*** is a user interface that provides graphical visualization of industrial monitoring and control system. HMI allows operators to view real time or near real time process information.



The Scenario– An industrial Hatchery System

- Hatchery System Model
- Security analyser scanned the Model
- Security Issue was identified
- Setter1PLC is vulnerable

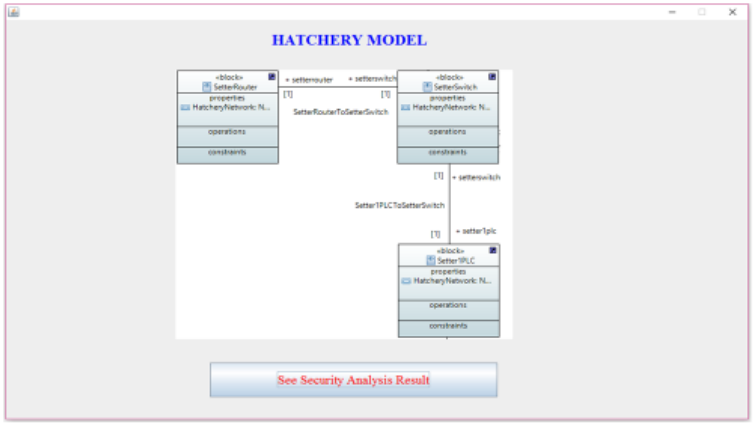
How will you solve the problem?



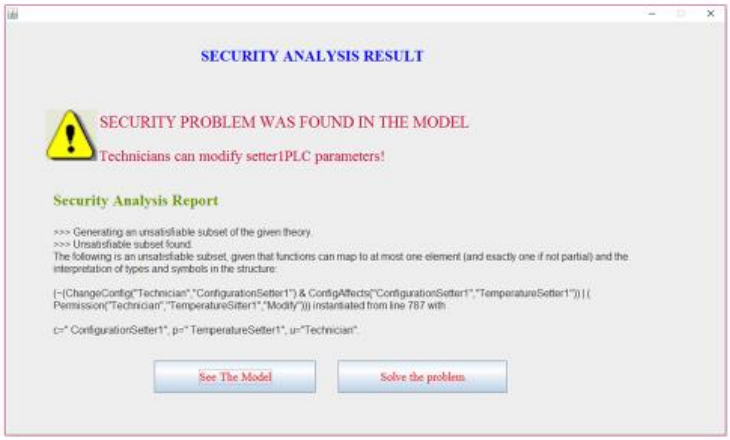
Finding a solution

You have 30 minute to find a possible solution

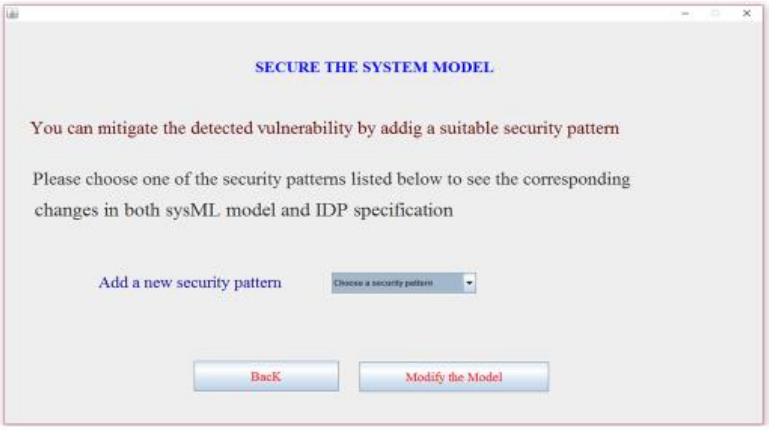
Applying the solution



Applying the solution

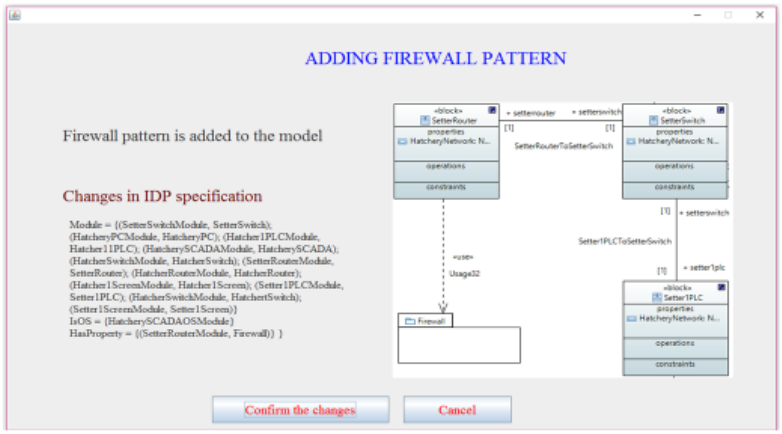


Applying the solution

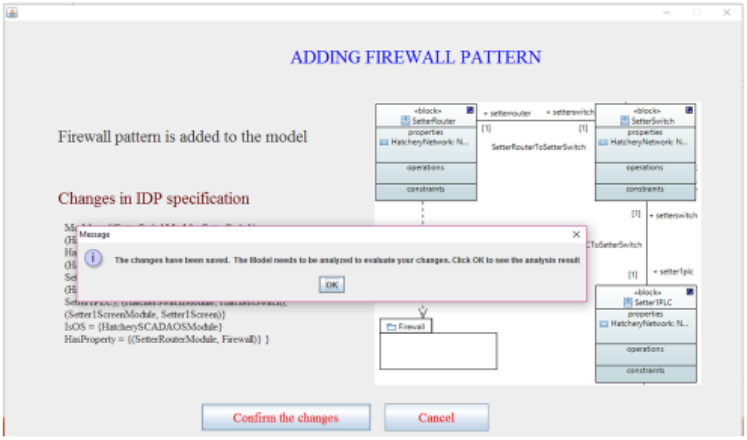


Applying the solution

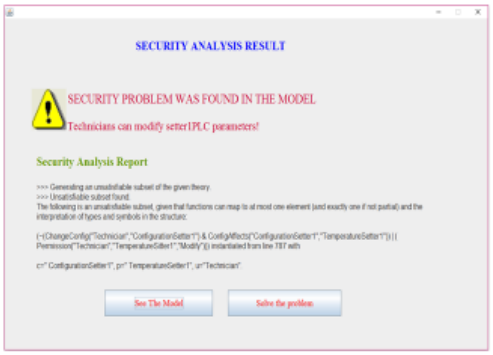
Applying the solution



Applying the solution



Analysis Result

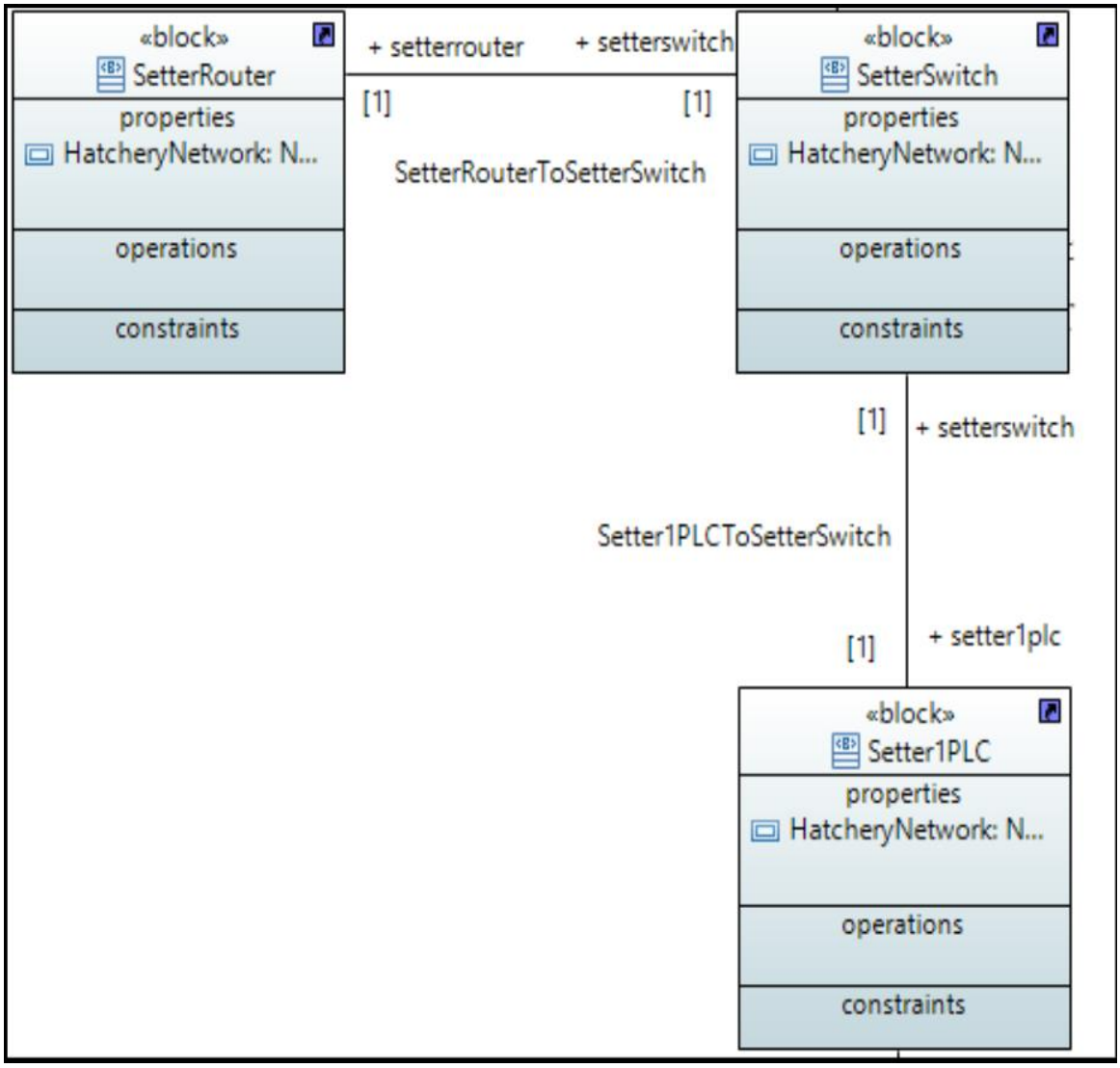


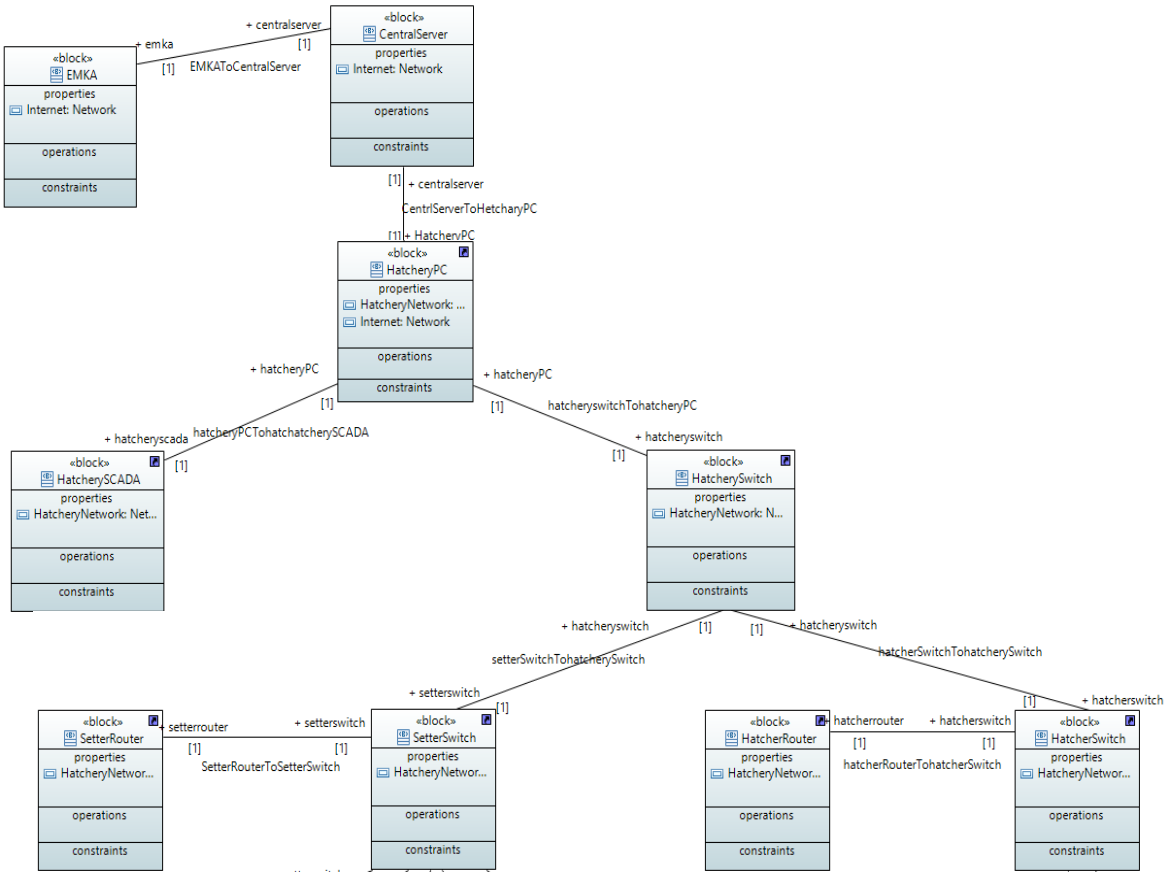
THANK YOU

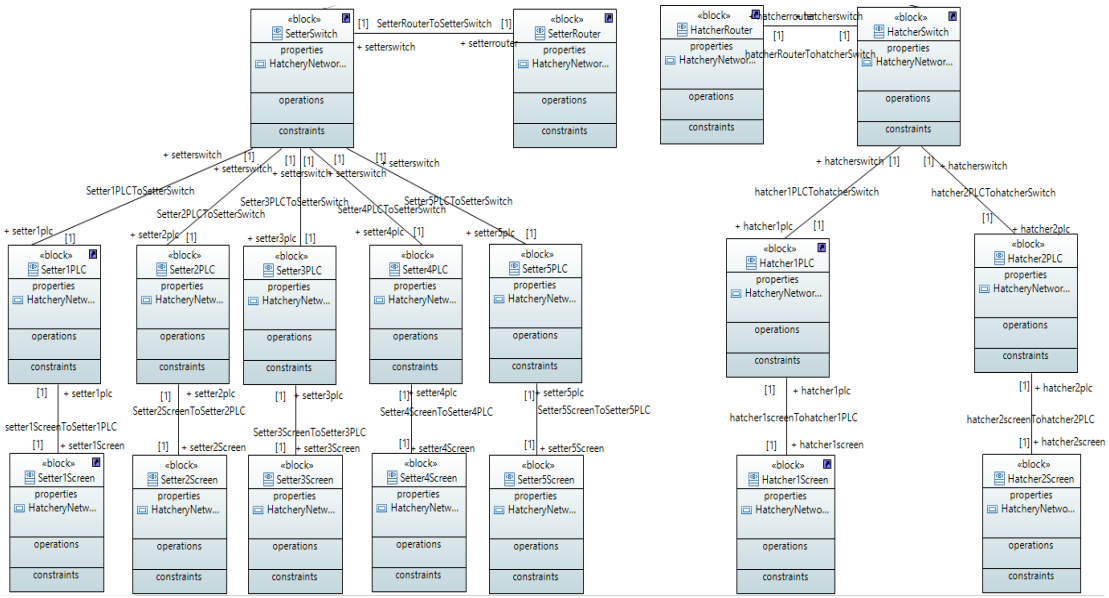
D.5 System Description

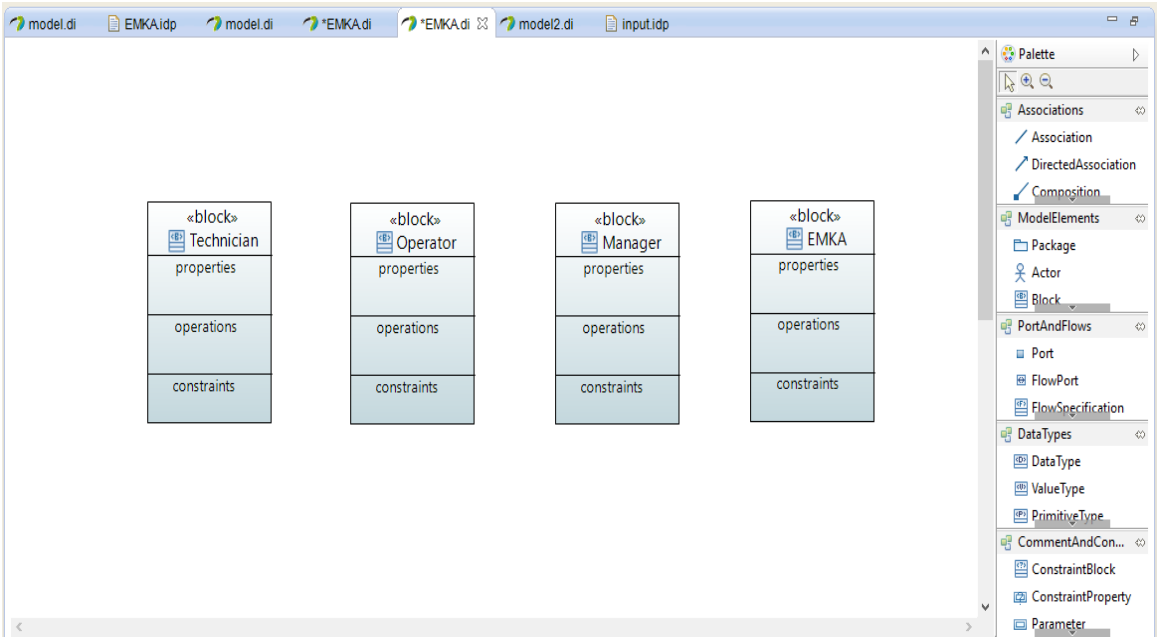
The system is an industrial hatchery. The hatchery consists of seven incubators, five setters and two hatchers. Each incubator can hold up to 115200 eggs. Eggs are initially put in one of the setter incubators, where they are turned hourly. Then they get transferred to the hatcher incubators to hatch. Each incubator consists of various sensors and actuators that are connected to a PLC (**Setter1PLC**,**Setter2PLC**, **Setter3PLC**, **Setter4PLC**, **Setter5PLC**, **Hatcher1PLC**, **Hatcher2PLC**). At the front of the incubator a touchscreen is used for monitoring and controlling the parameters (**Setter1Screen**, **Setter2Screen**, **Setter3Screen**, **Setter4Screen**, **Setter5Screen**, **Hatcher1Screen**, **Hatcher2Screen**). Each incubator room has a switch that all PLCs in that room are connected to (**SetterSwitch**, **HatcherSwitch**). This switch is connected to a wireless router (**SetterRouter**, **HatcherRouter**) which can be used for accessing the incubators with a mobile device, using an app that can take control of the touchscreens. The room switches are all connected to a switch in a centralized location (**HatcherySwitch**). In this location, you also find a server (**CentralServer**) that is used by the manufacturer to connect to the hatchery remotely. There is also an industrial PC (**HatcheryPC**) that logs all the data and can be used to control all incubators.

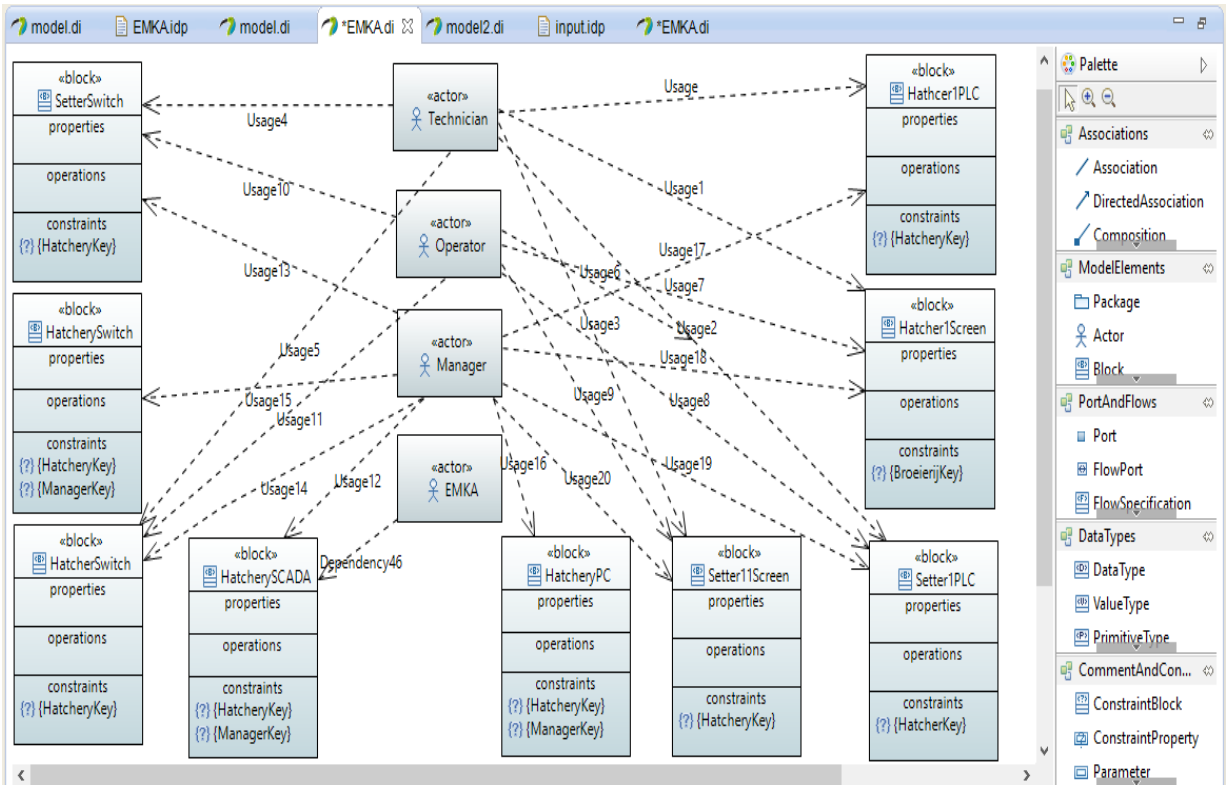
There are currently four different types of users in the hatchery. The least privileged users can look at the different parameters, but not change them. Additionally, they can reset the incubator alarms or turn off the sound. This is meant for technicians (Technician). The second lowest level is used by the operators (Operator) of the hatchery. This user group can change all parameters of the incubators, including the temperature settings, humidity, CO2 levels, etc. The local managers (Manager) make up the third level. They are able to do all the above, as well as change operator passwords, export data regarding login lists and alarms to USB, and so on. The highest level is reserved for the manufacturer (EMKA) of the incubators. When they log on remotely using their password, they also get access to additional information regarding errors and failures, so they can assist when problems occur.

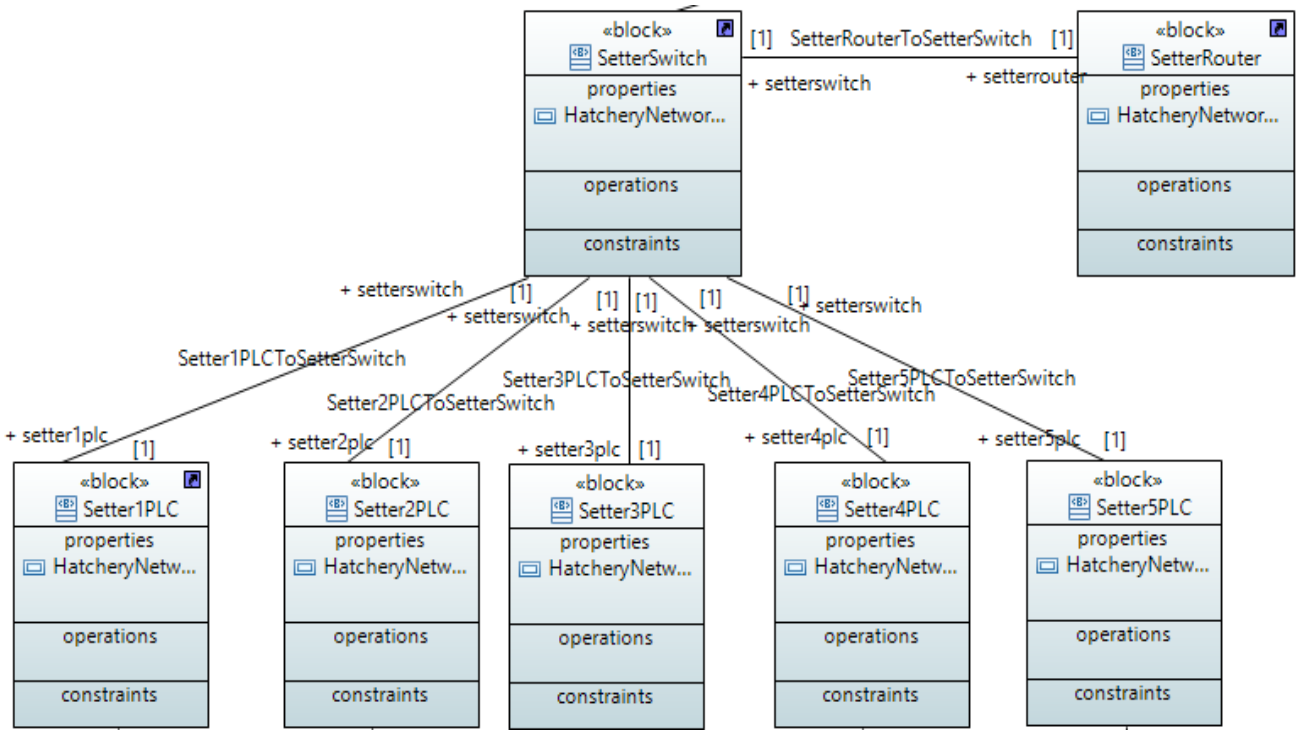

«block»
SetterRouter
properties
HatcheryNetwork: N...
operations
constraints
+ setterrouter
+ setterswitch
[1]
[1]
SetterRouterToSetterSwitch
«block»
SetterSwitch
properties
HatcheryNetwork: N...
operations
constraints
[1]
+ setterswitch
Setter1PLCToSetterSwitch
[1]
+ setter1plc
«block»
Setter1PLC
properties
HatcheryNetwork: N...
operations
constraints


«block»
EMKA
properties
Internet: Network
operations
constraints
+ emka
[1]
EMKAToCentralServer
+ centralserver
[1]
«block»
CentralServer
properties
Internet: Network
operations
constraints
[1]
+ centralserver
CentrlServerToHetcharyPC
[1]
+ HatcheryPC
«block»
HatcheryPC
properties
HatcheryNetwork: ...
Internet: Network
operations
constraints
+ hatcheryPC
[1]
+ hatcheryPC
[1]
hatcheryswitchTohatcheryPC
+ hatcheryscada
[1]
hatcheryPCTohatchatcherySCADA
+ hatcheryswitch
[1]
«block»
HatcherySCADA
properties
HatcheryNetwork: Net...
operations
constraints
«block»
HatcherySwitch
properties
HatcheryNetwork: N...
operations
constraints
+ hatcheryswitch
[1]
[1]
+ hatcheryswitch
setterSwitchTohatcherySwitch
hatcherSwitchTohatcherySwitch
+ setterswitch
[1]
+ hatcherswitch
«block»
SetterRouter
properties
HatcheryNetwor...
operations
constraints
+ setterrouter
[1]
+ setterswitch
[1]
SetterRouterToSetterSwitch
«block»
SetterSwitch
properties
HatcheryNetwor...
operations
constraints
«block»
HatcherRouter
properties
HatcheryNetwor...
operations
constraints
+ hatcherrouter
[1]
+ hatcherswitch
[1]
hatcherRouterTohatcherSwitch
«block»
HatcherSwitch
properties
HatcheryNetwor...
operations
constraints


«block»
SetterSwitch
properties
HatcheryNetwor...
operations
constraints
[1]
SetterRouterToSetterSwitch
[1]
«block»
SetterRouter
properties
HatcheryNetwor...
operations
constraints
+ setterswitch
+ setterrouter
«block»
HatcherRouter
properties
HatcheryNetwor...
operations
constraints
«block»
HatcherSwitch
properties
HatcheryNetwor...
operations
constraints
hatcherRouterTohatcherSwitch
+ setterswitch
+ hatcherswitch
Setter1PLCToSetterSwitch
Setter2PLCToSetterSwitch
Setter3PLCToSetterSwitch
Setter4PLCToSetterSwitch
Setter5PLCToSetterSwitch
hatcher1PLCTohatcherSwitch
hatcher2PLCTohatcherSwitch
+ setter1plc
+ setter2plc
+ setter3plc
+ setter4plc
+ setter5plc
+ hatcher1plc
+ hatcher2plc
«block»
Setter1PLC
«block»
Setter2PLC
«block»
Setter3PLC
«block»
Setter4PLC
«block»
Setter5PLC
«block»
Hatcher1PLC
«block»
Hatcher2PLC
setter1ScreenToSetter1PLC
Setter2ScreenToSetter2PLC
Setter3ScreenToSetter3PLC
Setter4ScreenToSetter4PLC
Setter5ScreenToSetter5PLC
hatcher1screenTohatcher1PLC
hatcher2screenTohatcher2PLC
+ setter1Screen
+ setter2Screen
+ setter3Screen
+ setter4Screen
+ setter5Screen
+ hatcher1screen
+ hatcher2screen
«block»
Setter1Screen
«block»
Setter2Screen
«block»
Setter3Screen
«block»
Setter4Screen
«block»
Setter5Screen
«block»
Hatcher1Screen
«block»
Hatcher2Screen


model.di
EMKA.idp
model.di
*EMKA.di
*EMKA.di
model2.di
input.idp
«block» Technician
properties
operations
constraints
«block» Operator
properties
operations
constraints
«block» Manager
properties
operations
constraints
«block» EMKA
properties
operations
constraints
Palette
Associations
Association
DirectedAssociation
Composition
ModelElements
Package
Actor
Block
PortAndFlows
Port
FlowPort
FlowSpecification
DataTypes
DataType
ValueType
PrimitiveType
CommentAndCon...
ConstraintBlock
ConstraintProperty
Parameter


model.di
EMKA.idp
model.di
*EMKA.di
model2.di
input.idp
*EMKA.di
«block» SetterSwitch
properties
operations
constraints
{?} {HatcheryKey}
«block» HatcherySwitch
properties
operations
constraints
{?} {HatcheryKey}
{?} {ManagerKey}
«block» HatcherSwitch
properties
operations
constraints
{?} {HatcheryKey}
«actor» Technician
«actor» Operator
«actor» Manager
«actor» EMKA
«block» HatcherySCADA
properties
operations
constraints
{?} {HatcheryKey}
{?} {ManagerKey}
«block» HatcheryPC
properties
operations
constraints
{?} {HatcheryKey}
{?} {ManagerKey}
«block» Setter11Screen
properties
operations
constraints
{?} {HatcheryKey}
«block» Hathcer1PLC
properties
operations
constraints
{?} {HatcheryKey}
«block» Hatcher1Screen
properties
operations
constraints
{?} {BroeierijKey}
«block» Setter1PLC
properties
operations
constraints
{?} {HatcherKey}
Usage
Usage4
Usage10
Usage13
Usage5
Usage15
Usage11
Usage14
Usage12
Dependency46
Usage1
Usage17
Usage6
Usage7
Usage3
Usage2
Usage18
Usage9
Usage8
Usage16
Usage20
Usage19
Palette
Associations
Association
DirectedAssociation
Composition
ModelElements
Package
Actor
Block
PortAndFlows
Port
FlowPort
FlowSpecification
DataTypes
DataType
ValueType
PrimitiveType
CommentAndCon...
ConstraintBlock
ConstraintProperty
Parameter


«block»
SetterSwitch
properties
HatcheryNetwor...
operations
constraints
[1] SetterRouterToSetterSwitch [1]
+ setterswitch
+ setterrouter
«block»
SetterRouter
properties
HatcheryNetwor...
operations
constraints
+ setterswitch [1]
Setter1PLCToSetterSwitch
+ setter1plc [1]
«block»
Setter1PLC
properties
HatcheryNetw...
operations
constraints
+ setterswitch [1]
Setter2PLCToSetterSwitch
+ setter2plc [1]
«block»
Setter2PLC
properties
HatcheryNetw...
operations
constraints
[1] [1]
Setter3PLCToSetterSwitch
+ setterswitch
+ setter3plc [1]
«block»
Setter3PLC
properties
HatcheryNetw...
operations
constraints
[1] [1]
Setter4PLCToSetterSwitch
+ setterswitch
+ setter4plc [1]
«block»
Setter4PLC
properties
HatcheryNetw...
operations
constraints
[1] [1]
Setter5PLCToSetterSwitch
+ setterswitch
+ setter5plc [1]
«block»
Setter5PLC
properties
HatcheryNetw...
operations
constraints

References

AAMODT, A. Case-based reasoning-an introduction. A newforce in advanced systems development, Unicom Seminars, 1995. 9-23.

AAMODT, A. & PLAZA, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications,* 7**,** 39-59.

ABOUZAKHAR, N. Critical Infrastructure Cybersecurity: A Review of Recent Threats and Violations. European Conference on Cyber Warfare and Security, 2013. Academic Conferences International Limited, 1.

ABRAMS, M. & WEISS, J. 2008. Malicious control system cyber security attack case study–Maroochy Water Services, Australia. *McLean, VA: The MITRE Corporation*.

AL-DEBEI, M. M. & FITZGERALD, G. 2009. OntoEng: a design method for ontology engineering in information systems. *Proceedings of the ACM OOPSLA'09, ODiSE,* 2009.

ALCARAZ, C., FERNANDEZ, G. & CARVAJAL, F. 2012. Security aspects of SCADA and DCS environments. *Critical Infrastructure Protection.* Springer.

ALLEN JR, J. C. 2011. Sample Size Calculation for Two Independent Groups: A Useful Rule of Thumb. *Proceedings of Singapore Healthcare,* 20**,** 138-140.

AMAECHI, A. & COUNSELL, S. Challenges and issues in the system architecting for systemic risk infrastructure system: An industrial case study. Complex Systems (ICCS), 2012 International Conference on, 2012. IEEE, 1-8.

ANNEX, V. 2011. Protecting Industrial Control Systems.

ANWAR, Z. & MALIK, A. W. 2014. Can a DDoS attack meltdown my data center? A simulation study and defense strategies. *IEEE Communications Letters,* 18**,** 1175-1178.

ARJONA, M., RUIZ, J. F. & MAÑA, A. Security Patterns for Local Assurance in Cloud Applications. International Workshop on Engineering Cyber Security and Resilience ECSaR'14, 2014.

AVISON, D. E., DWIVEDI, Y. K., FITZGERALD, G. & POWELL, P. 2008. The beginnings of a new era: time to reflect on 17 years of the ISJ. *Information Systems Journal,* 18**,** 5-21.

AXELROD, C. W. Applying lessons from safety-critical systems to security-critical software. Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island, 2011. IEEE, 1-6.

BAKER, S. E., EDWARDS, R. & DOIDGE, M. 2012. How many qualitative interviews is enough?: Expert voices and early career reflections on sampling and cases in qualitative research.

BARROWS, H. S. & TAMBLYN, R. M. 1980. *Problem-based learning: An approach to medical education*, Springer Publishing Company.

BASILI, V. R. 2007. The role of controlled experiments in software engineering research. *Empirical Software Engineering Issues. Critical Assessment and Future Directions.* Springer.

BASSIL, Y. 2012. A simulation model for the waterfall software development life cycle. *arXiv preprint arXiv:1205.6904*.

BERGMANN, R., KOLODNER, J. & PLAZA, E. 2005. Representation in case-based reasoning. *The Knowledge Engineering Review,* 20**,** 209-213.

BERNARDO, A., LANDICHO, A. & LAGUADOR, J. M. 2014. On-the-Job Training Performance of Students from AB Paralegal Studies for SY 2013-2014. *Studies in Social Sciences and Humanities,* 1**,** 122-129.

BERNSTEIN, R. J. 2011. *Beyond objectivism and relativism: Science, hermeneutics, and praxis*, University of Pennsylvania Press.

BISCHOFBERGER, W. R. & POMBERGER, G. 2012. *Prototyping-oriented software development: Concepts and tools*, Springer Science & Business Media.

BLACKWELL, A. F. & GREEN, T. R. A Cognitive Dimensions questionnaire optimised for users. Proceedings of the Twelfth Annual Meeting of the Psychology of Programming Interest Group, 2000. 137-152.

BOUD, D. & ROONEY, D. 2015. What can higher education learn from the workplace? *Transformative perspectives and processes in higher education.* Springer.

BOYES, H. Best Practices in an ICS Environment. IET Conference Proceedings, 2015. The Institution of Engineering & Technology.

BRÄNDLE, M. & NAEDELE, M. 2008. Security for process control systems: An overview. *IEEE Security & Privacy*, 6**,** 24-29.

BRAUN, V. & CLARKE, V. 2006. Using thematic analysis in psychology. *Qualitative research in psychology,* 3**,** 77-101.

BUNKE, M., KOSCHKE, R. & SOHR, K. 2012. Organizing security patterns related to security and pattern recognition requirements. *International Journal on Advances in Security,* 5.

BYRES, E. J., HOFFMAN, D. & KUBE, N. 2006. On shaky ground–A study of security vulnerabilities in control protocols. *Proc. 5th American Nuclear Society Int. Mtg. on Nuclear Plant Instrumentation, Controls, and HMI Technology*.

CAMACHO, D., ORTIGOSA, A., PULIDO, E. & R-MORENO, M. D. 2008. AI techniques for monitoring student learning process. *Information Science Reference, formerly Idea Group Publishing, Ed. by Francisco J. Garcıa*.

CAWLEY, P. 1989. The introduction of a problem-based option into a conventional engineering degree course. *Studies in Higher Education,* 14**,** 83-95.

CHEMINOD, M., DURANTE, L. & VALENZANO, A. 2013. Review of security issues in industrial networks. *IEEE Transactions on Industrial Informatics,* 9**,** 277-293.

CHIEN, E., O'MURCHU, L. & FALLIERE, N. W32. Duqu: The Precursor to the Next Stuxnet. LEET, 2012.

CHUA, W. F. 1986. Radical developments in accounting thought. *Accounting review***,** 601-632.

CLEVEN, A., GUBLER, P. & HÜNER, K. M. Design alternatives for the evaluation of design science research artifacts. Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, 2009. ACM, 19.

COHEN, L., MANION, L. & MORRISON, K. 2013. *Research methods in education*, Routledge.

COLLINS, S. & MCCOMBIE, S. 2012. Stuxnet: the emergence of a new cyber weapon and its implications. *Journal of Policing, Intelligence and Counter Terrorism,* 7**,** 80-91.

COLLIS, J. & HUSSEY, R. 2013. *Business research: A practical guide for undergraduate and postgraduate students*, Palgrave macmillan.

COOLICAN, H. 2014. *Research methods and statistics in psychology*, Psychology Press.

CPNI. 2016. *CPNI* [Online]. Available: https://www.cpni.gov.uk/ [Accessed 14/10 2016].

CREATORS, W. S. S. 2013. To Kill a Centrifuge.

CRESWELL, J. W. 2012. *Qualitative inquiry and research design: Choosing among five approaches*, Sage publications.

CROUCH, M. & MCKENZIE, H. 2006. The logic of small samples in interview-based qualitative research. *Social science information,* 45**,** 483-499.

CYBERATTACKS, G. E. 2011. Night dragon. *McAfee Foundstone Professional Services and McAfee Labs*.

DA SILVA JÚNIOR, L. S., GUÉHÉNEUC, Y.-G. & MULLINS, J. 2013. An approach to formalise security patterns. Tech. rep., École Polytechnique de Montréal, Montréal Quebéc.

DENZIN, N. K. & LINCOLN, Y. S. 2011. *The Sage handbook of qualitative research*, Sage.

DICICCO-BLOOM, B. & CRABTREE, B. F. 2006. The qualitative research interview. *Medical education,* 40**,** 314-321.

DIMITROV, D. M. & RUMRILL JR, P. D. 2003. Pretest-posttest designs and measurement of change. *Work,* 20**,** 159-165.

DONALDSON, R. A good start in architecture. Problem-based learning: The Newcastle workshop, 1989. University of Newcastle Newcastle, Australia, 41-53.

DRIAS, Z., SERHROUCHNI, A. & VOGEL, O. Analysis of cyber security for industrial control systems. Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), 2015 International Conference on, 2015. IEEE, 1-8.

DURRANI, S., JATTALA, I., FAROOQI, J., SHAKEEL, N. & MURAD, M. Design and development of wireless RTU and cybersecurity framework for SCADA system. Information & Communication Technologies (ICICT), 2013 5th International Conference on, 2013. IEEE, 1-6.

DYBÅ, T., KAMPENES, V. B. & SJØBERG, D. I. 2006. A systematic review of statistical power in software engineering experiments. *Information and Software Technology,* 48**,** 745-755.

FAN, X., FAN, K., WANG, Y. & ZHOU, R. Overview of cyber-security of industrial control system. Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), 2015 International Conference on, 2015. IEEE, 1-7.

FARRELL, A. 2007. Selecting a Software Development Methodology based on Organizational Characteristics. *An Essay Submitted in Partial Fulfillment of the Requirements for the Degree of "Master of Science in Information Systems", Athabasca University, Athabasca*.

FAY, B. 1987. An alternative view: Interpretive social science. *Interpreting*.

FERNANDEZ-BUGLIONI, E. 2013. *Security patterns in practice: designing secure architectures using software patterns*, John Wiley & Sons.

FERNÁNDEZ-LÓPEZ, M. & GÓMEZ-PÉREZ, A. 2002. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, 17**,** 129-156.

FERNANDEZ, E. B., WASHIZAKI, H. & YOSHIOKA, N. Abstract security patterns. Proceedings of the 15th Conference on Pattern Languages of Programs, 2008. ACM, 4.

FERNANDEZ, E. B., YOSHIOKA, N., WASHIZAKI, H., JÜRJENS, J., VANHILST, M. & PERNUL, G. 2011. Using security patterns to develop secure systems. *IGI Global***,** 16-31.

FLEURY, T., KHURANA, H. & WELCH, V. Towards a taxonomy of attacks against energy control systems. International Conference on Critical Infrastructure Protection, 2008. Springer, 71-85.

FOO, E., BRANAGAN, M. & MORRIS, T. A proposed australian industrial control system security curriculum. System Sciences (HICSS), 2013 46th Hawaii International Conference on, 2013. IEEE, 1754-1762.

FOVINO, I. N., MASERA, M., GUGLIELMI, M., CARCANO, A. & TROMBETTA, A. Distributed intrusion detection system for SCADA protocols. International Conference on Critical Infrastructure Protection, 2010. Springer, 95-110.

FRANCIA III, G. A. Critical infrastructure security curriculum modules. Proceedings of the 2011 Information Security Curriculum Development Conference, 2011. ACM, 54-58.

FRIESEN, N. 2005. Interoperability and learning objects: An overview of e-learning standardization. *Interdisciplinary Journal of Knowledge and Learning Objects*, 1**,** 23-31.

GALLIERS, R. 1992. *Information systems research: Issues, methods and practical guidelines*, Blackwell Scientific.

GALLOWAY, B. & HANCKE, G. P. 2013. Introduction to industrial control networks. *IEEE Communications Surveys and Tutorials*, 15**,** 860-880.

GAMMA, E. 1995. *Design patterns: elements of reusable object-oriented software*, Pearson Education India.

GARRIDO, A., MORALES, L. & SERINA, I. Applying Case-Based Planning to Personalized E-learning. DMS, 2011. Citeseer, 228-233.

GARRIDO, A., MORALES, L. & SERINA, I. Using AI Planning to Enhance E-Learning Processes. ICAPS, 2012.

GARRIDO, A., ONAINDIA, E., MORALES, L., CASTILLO, L., FERNÁNDEZ, S. & BORRAJO, D. 2009. Modeling E-Learning Activities in Automated Planning∗.

GIBBONS, M. T. 1987. Introduction: The politics of interpretation. *Interpreting politics***,** 1-31.

GRAHAM, J., HIEB, J. & NABER, J. Improving cybersecurity for Industrial Control Systems. Industrial Electronics (ISIE), 2016 IEEE 25th International Symposium on, 2016. IEEE, 618-623.

GREEN, T. R. G. & PETRE, M. 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing,* 7**,** 131-174.

GREGOR, S. & JONES, D. 2007. The anatomy of a design theory. *Journal of the Association for Information Systems,* 8**,** 312.

GRIMM, T. 2004. *User's guide to rapid prototyping,* Society of Manufacturing Engineers.

GROBAUER, B., WALLOSCHEK, T. & STOCKER, E. 2011. Understanding cloud computing vulnerabilities. *IEEE Security & Privacy,* 9**,** 50-57.

GUEST, G., BUNCE, A. & JOHNSON, L. 2006. How many interviews are enough? An experiment with data saturation and variability. *Field methods,* 18**,** 59-82.

HADZIOSMANOVIC, D., BOLZONI, D., ETALLE, S. & HARTEL, P. Challenges and opportunities in securing industrial control systems. Complexity in Engineering (COMPENG), 2012, 2012. IEEE, 1-6.

HALKIER, B. & JENSEN, I. 2011. Methodological challenges in using practice theory in consumption research. Examples from a study on handling nutritional contestations of food consumption. *Journal of Consumer Culture,* 11**,** 101-123.

HAMID, B., GÜRGENS, S. & FUCHS, A. 2016. Security patterns modeling and formalization for pattern-based development of secure software systems. *Innovations in Systems and Software Engineering,* 12**,** 109-140.

HANID, M. 2014. *Design science research as an approach to develop conceptual solutions for improving cost management in construction.* University of Salford.

HASAN, L., MORRIS, A. & PROBETS, S. 2009. Using Google Analytics to evaluate the usability of e-commerce sites. *Human centered design***,** 697-706.

HASHEMINEJAD, S. M. H. & JALILI, S. Selecting proper security patterns using text classification. Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on, 2009. IEEE, 1-5.

HE, H., MAPLE, C., WATSON, T., TIWARI, A., MEHNEN, J., JIN, Y. & GABRYS, B. The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational

intelligence. Evolutionary Computation (CEC), 2016 IEEE Congress on, 2016. IEEE, 1015-1021.

HENTEA, M. 2008. Improving security for SCADA control systems. *Interdisciplinary Journal of Information, Knowledge, and Management,* 3**,** 73-86.

HEVNER, A. & CHATTERJEE, S. 2010. *Design science research in information systems*, Springer.

HODGINS, W. & DUVAL, E. 2002. Draft standard for learning object metadata. *IEEE,* 1484**,** 1-2002.

HOWARD, M. & LEBLANC, D. 2003. Writing Secure Code 2ndEd. *Microsoft Press*.

ICS-CERT. *Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies [Online]* [Online]. Available: https://ics-cert.us-cert.gov/ICS-CERT-releases-Recommended-Practice-Improving-Industrial-Control-System-Cybersecurity-Defense [Accessed].

ISMAIL, S., SITNIKOVA, E. & SLAY, J. Using integrated system theory approach to assess security for SCADA systems cyber security for critical infrastructures: A pilot study. Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on, 2014. IEEE, 1000-1006.

JACOBS, R. 2003. *Structured on-the-job training: Unleashing employee expertise in the workplace*, Berrett-Koehler Publishers.

JOHNSON, M. 2004. Personalised learning: an emperor's outfit. *Institute for Public Policy Research (IPPR), London*.

JONASSEN, D. H. & HUNG, W. 2008. All problems are not equal: Implications for problem-based learning. *Interdisciplinary Journal of Problem-Based Learning,* 2**,** 4.

JUGDER, N. 2014. *The influence of the market on curricular provision by higher education institutions in Mongolia*, University of Leeds.

KARGL, F., VAN DER HEIJDEN, R. W., KÖNIG, H., VALDES, A. & DACIER, M. C. 2014. Insights on the security and dependability of industrial control systems. *IEEE security & privacy,* 12**,** 75-78.

KEELE, S. 2007. Guidelines for performing systematic literature reviews in software engineering. *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*. sn.

KHAZANCHI, D. & MUNKVOLD, B. E. On the rhetoric and relevance of IS research paradigms: a conceptual framework and some propositions. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, 2003. IEEE, 10 pp.

KITCHENHAM, B., BRERETON, O. P., BUDGEN, D., TURNER, M., BAILEY, J. & LINKMAN, S. 2009. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology,* 51**,** 7-15.

KLEIN, H. K. & MYERS, M. D. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS quarterly***,** 67-93.

KO, A. J., LATOZA, T. D. & BURNETT, M. M. 2015. A practical guide to controlled experiments of software engineering tools with human participants. *Empirical Software Engineering,* 20**,** 110-141.

KOVALCHICK, A. & DAWSON, K. 2004. *Education and technology: An encyclopedia*, Abc-clio.

KRESIMIR, R., MARIJANA, B. G. & VLADO, M. 2014. Development of the Intelligent System for the use of University Information System. *Procedia Engineering,* 69**,** 402-409.

KROTOFIL, M. & GOLLMANN, D. Industrial control systems security: What is happening? Industrial Informatics (INDIN), 2013 11th IEEE International Conference on, 2013. IEEE, 670-675.

KUANG, C., MIAO, Q. & CHEN, H. 2006. Analysis of software vulnerability. *WSEAS Transactions on Computers Research,* 1**,** 45.

KUMAR, S. & PHROMMATHED, P. 2005. *Research methodology*, Springer.

KUNSMAN, S., BRAENDLE, M., DE WIJS, B. & HOHLBAUM, F. Texas A&M university 68th annual conference for protective relay engineers replacing fear with knowledge-cyber security for substation automation, protection and control systems. Protective Relay Engineers, 2015 68th Annual Conference for, 2015. IEEE, 608-621.

KURSCHEID, J. Joining technical and organizational measures to secure process IT in critical infrastructure. Security in Critical Infrastructures Today, Proceedings of International ETG-Congress 2013; Symposium 1:, 2013. VDE, 1-6.

KUTAR, M., BRITTON, C. & BARKER, T. A comparison of empirical study and cognitive dimensions analysis in the evaluation of UML diagrams. Procs of the 14th Workshop of the Psychology of Programming Interest Group (PPIG 14), 2002.

KUZEL, A. J. 1992. Sampling in qualitative inquiry.

LAFOREST, J. 2009. Safety diagnosis tool kit for local communities. *Guide to organizing semi-structured interviews with key informants,* 11.

LANGNER, R. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy,* 9**,** 49-51.

LARKIN, R. D., LOPEZ JR, J., BUTTS, J. W. & GRIMAILA, M. R. 2014. Evaluation of security solutions in the SCADA environment. *ACM SIGMIS Database,* 45**,** 38-53.

LATOZA, T. D. & MYERS, B. A. Designing useful tools for developers. Proceedings of the 3rd ACM SIGPLAN workshop on Evaluation and usability of programming languages and tools, 2011. ACM, 45-50.

LEMAIRE, L., LAPON, J., DE DECKER, B. & NAESSENS, V. A SysML extension for security analysis of industrial control systems. Proceedings of the 2nd International Symposium on ICS & SCADA Cyber Security Research 2014, 2014. BCS, 1-9.

LEMAIRE, L., VOSSAERT, J., JANSEN, J. & NAESSENS, V. Extracting vulnerabilities in industrial control systems using a knowledge-based system. Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research, 2015. British Computer Society, 1-10.

LITZINGER, T., LATTUCA, L. R., HADGRAFT, R. & NEWSTETTER, W. 2011. Engineering education and the development of expertise. *Journal of Engineering Education,* 100**,** 123-150.

LUALLEN, M. E. & LABRUYERE, J.-P. Developing a critical infrastructure and control systems cybersecurity curriculum. System Sciences (HICSS), 2013 46th Hawaii International Conference on, 2013. IEEE, 1782-1791.

LUIIJF, E. Cyber (In-) security of Industrial Control Systems: A Societal Challenge. International Conference on Computer Safety, Reliability, and Security, 2015. Springer, 7-15.

LUND, A. M. 2001. Measuring Usability with the USE Questionnaire12.". *Usability interface,* 8**,** 3-6.

MAITLAND, B. 1991. Problem-based learning for an architecture degree. *The challenge of problem-based learning*.

MAÑA, A., FERNANDEZ, E. B., RUIZ, J. F. & RUDOLPH, C. Towards computer-oriented security patterns. Proceedings of the 20th Conference on Pattern Languages of Programs, 2013. The Hillside Group, 13.

MARCH, S. T. & SMITH, G. F. 1995. Design and natural science research on information technology. *Decision support systems,* 15**,** 251-266.

MASOOD, R. 2016. Assessment of Cyber Security Challenges in Nuclear Power Plants Security Incidents, Threats, and Initiatives.

MAVROMMATIS, G. 2008. Learning objects and objectives towards automatic learning construction. *European Journal of Operational Research,* 187**,** 1449-1458.

MCGREW, R. W. & VAUGHN, R. B. 2009. Discovering vulnerabilities in control system human–machine interface software. *Journal of Systems and Software,* 82**,** 583-589.

MILES, M. B. & HUBERMAN, A. M. 1994. Qualitative data analysis: A sourcebook. *Beverly Hills: Sage Publications*.

MILLS, J. E. & TREAGUST, D. F. 2003. Engineering education—Is problem-based or project-based learning the answer. *Australasian journal of engineering education,* 3**,** 2-16.

MINGERS, J. 2001. Combining IS research methods: towards a pluralist methodology. *Information systems research,* 12**,** 240-259.

MIYACHI, T. & YAMADA, T. Current issues and challenges on cyber security for industrial automation and control systems. SICE Annual Conference (SICE), 2014 Proceedings of the, 2014. IEEE, 821-826.

MOLENDA, M. 2003. In search of the elusive ADDIE model. *Performance improvement,* 42**,** 34-37.

MORSE, J. M. 1994. Designing funded qualitative research.

MOTII, A., HAMID, B., LANUSSE, A. & BRUEL, J.-M. Guiding the selection of security patterns based on security requirements and pattern classification. Proceedings of the 20th European Conference on Pattern Languages of Programs, 2015. ACM, 10.

MOURATIDIS, H. 2006. *Integrating Security and Software Engineering: Advances and Future Visions: Advances and Future Visions*, Igi Global.

MYERS, M. D. 1997. Qualitative research in information systems. *Management Information Systems Quarterly,* 21**,** 241-242.

NAVARRO, D., MÉNDEZ, J. C., BERRÍOS, K., ORTIZ-RIVERA, E. & ARZUAGA, E. Using cybersecurity as an engineering education approach on computer engineering to learn about Smart Grid technologies and the next generation of electric power systems. Frontiers in Education Conference (FIE), 2014 IEEE, 2014. IEEE, 1-8.

NELSO, T. & CHAFFIN, M. 2011. Common cybersecurity vulnerabilities in industrial control systems. *Control Systems Security Program*.

NERC. 2016. *North American Electric Reliability Corporation (NERC)* [Online]. Available: http://www.nerc.com/Pages/default.aspx [Accessed 20/10 2016].

NGUYEN, P. H. 2015. Model-Driven Security based on A Unified System of Security Design Patterns. University of Luxembourg.

NGUYEN, P. H., KLEIN, J. & LE TRAON, Y. Model-driven security with a system of aspect-oriented security design patterns. Proceedings of the 2nd Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling, 2014. ACM, 51.

NORMAN, G. R. & SCHMIDT, H. G. 2016. Revisiting 'Effectiveness of problem-based learning curricula: theory, practice and paper darts'. *Medical education,* 50**,** 793-797.

NOVAK, T. & TREYTL, A. Functional safety and system security in automation systems-a life cycle model. Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, 2008. IEEE, 311-318.

NSTB. 2016. *National Transportation Safety Board* [Online]. Available: http://www.ntsb.gov/about/Pages/default.aspx [Accessed 10/11 2016].

OATES, B. J. 2005. *Researching information systems and computing*, Sage.

OBREGON, L. 2015. Secure architecture for industrial control systems. *SANS Institute InfoSec Reading Room*.

OFFERMANN, P., LEVINA, O., SCHÖNHERR, M. & BUB, U. Outline of a design science research process. Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, 2009. ACM, 7.

ONAINDIA, E., SAPENA, O. & GARRIDO, A. 2007. An AI planning-based approach for automated design of learning routes. *ECEL2007-Proceedings of the European Confernce on e-Learning 2007: ECEL2007***,** 453.

ORLIKOWSKI, W. J. & BAROUDI, J. J. 1991. Studying information technology in organizations: Research approaches and assumptions. *Information systems research,* 2**,** 1-28.

OZMENT, J. A. 2007. *Vulnerability discovery & software security.* University of Cambridge.

PATTON, M. Q. 1990. *Qualitative evaluation and research methods*, SAGE Publications, inc.

PAUNA, A., LINARES, S., PAREDES, I., VALIENTE, J., PILAR, J., BRUNA, T., MARTÍNEZ, S. & HUISTRA, A. 2014. Certification of Cyber Security skills of ICS/SCADA professionals. *The European Union Agency for Network and Information Security (ENISA), Heraklion*.

PEACHEY, D. R. & MCCALLA, G. I. 1986. Using planning techniques in intelligent tutoring systems. *International Journal of Man-Machine Studies,* 24**,** 77-98.

PEDROZA, G., APVRILLE, L. & KNORRECK, D. Avatar: A sysml environment for the formal verification of safety and security properties. New Technologies of Distributed Systems (NOTERE), 2011 11th Annual International Conference on, 2011. IEEE, 1-10.

PERMANN, M., HAMMER, J., LEE, K. & ROHDE, K. Mitigations for security vulnerabilities found in control system networks. Proceedings of the 16th Annual Joint ISA POWID/EPRI Controls and Instrumentation Conference, 2006.

PERRENET, J., BOUHUIJS, P. & SMITS, J. 2000. The suitability of problem-based learning for engineering education: theory and practice. *Teaching in higher education,* 5**,** 345-358.

PILKINGTON, N. T., LI, J. & XIE, F. Eside: An integrated development environment for component-based embedded systems. Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International, 2009. IEEE, 305-314.

POLSANI, P. R. 2006. Use and abuse of reusable learning objects. *Journal of Digital information,* 3.

POULSEN, K. 2003. Slammer worm crashed Ohio nuke plant network. Security Focus. *SecurityFocus. com*.

PREZ, J. & MACHNICKI, D. 2013. ValueSec D5. 3-Description of developed tools and data. ValueSec.

PURAO, S. 2002. Design research in the technology of information systems: Truth or dare. *GSU Department of CIS Working Paper***,** 45-77.

RADATZ, J., GERACI, A. & KATKI, F. 1990. IEEE standard glossary of software engineering terminology. *IEEE Std,* 610121990**,** 3.

ROONEY, D., WILLEY, K., GARDNER, A., BOUD, D., REICH, A., FITZGERALD, T., WILLIAMS, B., FIGUEIREDO, J. & TREVELYAN, J. 2014. Engineers' professional learning: Through the lens of practice. *Engineering practice in a global context: understanding the technical and the social***,** 265-280.

ROSSMAN, G. B. & MARSHALL, C. 1995. Designing qualitative research.

ROTHWELL, W. J. & KAZANAS, H. C. 2004. *Improving on-the-job training: How to establish and operate a comprehensive OJT program*, John Wiley & Sons.

SADEGHI, A.-R., WACHSMANN, C. & WAIDNER, M. Security and privacy challenges in industrial internet of things. Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE, 2015. IEEE, 1-6.

SAJID, A., ABBAS, H. & SALEEM, K. 2016. Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges. *IEEE Access*, 4**,** 1375-1384.

SAKS, A. M. & BURKE-SMALLEY, L. A. 2014. Is transfer of training related to firm performance? *International Journal of Training and Development*, 18**,** 104-115.

SALDAÑA, J. 2015. *The coding manual for qualitative researchers*, Sage.

SANS. 2016. *SANS* [Online]. Available: https://www.sans.org/ [Accessed 17/10 2016].

SAVOLA, R. & AHONEN, P. Information security challenges in industrial automation systems. Industrial Informatics, 2006 IEEE International Conference on, 2006. IEEE, 581-586.

SCADAHACKER. 2016. *SCADAHACKER* [Online]. Available: WWW.SCADAhacker.COM [Accessed 20/10 2016].

SCHMIDT, D. C. & MCCORMICK, Z. Producing and delivering a coursera MOOC on pattern-oriented software architecture for concurrent and networked software. Proceedings of the 2013 companion publication for conference on Systems, programming, & applications: software for humanity, 2013. ACM, 167-176.

SCHUMACHER, M., FERNANDEZ-BUGLIONI, E., HYBERTSON, D., BUSCHMANN, F. & SOMMERLAD, P. 2013. *Security Patterns: Integrating security and systems engineering*, John Wiley & Sons.

SCOTT, W. R. 2013. *Institutions and organizations: Ideas, interests, and identities*, Sage Publications.

SEBBA, J., BROWN, N., STEWARD, S., GALTON, M. & JAMES, M. 2007. An investigation of personalised learning approaches used by schools. *Nottingham: DfES Publications*.

SETTE, F. M., VAQUERO, T. S., PARK, S. W. & SILVA, J. R. 2008. Are Automated Planners up to Solve Real Problems? *IFAC Proceedings Volumes*, 41**,** 15817-15824.

SHADISH, W. R., COOK, T. D. & CAMPBELL, D. T. 2002. *Experimental and quasi-experimental designs for generalized causal inference*, Wadsworth Cengage learning.

SHUKLA, S. K. Cyber Security of Cyber Physical Systems: Cyber Threats and Defense of Critical Infrastructures. VLSI Design and 2016 15th International

Conference on Embedded Systems (VLSID), 2016 29th International Conference on, 2016. IEEE, 30-31.

SIMON, H. A. 1996. *The sciences of the artificial*, MIT press.

SJØBERG, D. I., HANNAY, J. E., HANSEN, O., KAMPENES, V. B., KARAHASANOVIC, A., LIBORG, N.-K. & REKDAL, A. C. 2005. A survey of controlled experiments in software engineering. *IEEE transactions on software engineering,* 31**,** 733-753.

SPSS 2013. *IBM Corp*.

STEVEN, J. 2006. Adopting an enterprise software security framework. *IEEE Security & Privacy,* 4**,** 84-87.

STONER, J. A. 2004. Cross-over trials in clinical research. Taylor & Francis.

STOUFFER, K., FALCO, J. & SCARFONE, K. 2011. Guide to industrial control systems (ICS) security. *NIST special publication,* 800**,** 16-16.

TOTH, P. & KLEIN, P. 2013. A role-based model for federal information technology/cyber security training. *NIST special publication,* 800**,** 16.

TRIFFITT, E. & KHAZAEI, B. A study of usability of Z formalism based on cognitive dimensions. Proceedings of the 14 th Annual Meeting of the Psychology of Programming Interest Group (PPIG, 2002. Citeseer.

UDO, M., VAQUERO, T. S., SILVA, J. R. & TONIDANDEL, F. Lean software development domain. Proceedings of ICAPS 2008 Scheduling and Planning Application workshop. Sydney, Australia, 2008.

UR-REHMAN, O. & ZIVIC, N. Secure design patterns for security in smart metering systems. Modelling Symposium (EMS), 2015 IEEE European, 2015. IEEE, 278-283.

VAISHNAVI, V. & KUECHLER, W. 2004. Design research in information systems.

VAISHNAVI, V. K. & KUECHLER, W. 2015. *Design science research methods and patterns: innovating information and communication technology,* Crc Press.

VAISMORADI, M., TURUNEN, H. & BONDAS, T. 2013. Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & health sciences,* 15**,** 398-405.

VAQUERO, T. S., ROMERO, V., TONIDANDEL, F. & SILVA, J. R. itSIMPLE 2.0: An Integrated Tool for Designing Planning Domains. ICAPS, 2007. 336-343.

VAQUERO, T. S., TONIDANDEL, F., DE BARROS, L. N. & SILVA, J. R. On the Use of UML. P for Modeling a Real Application as a Planning Problem. ICAPS, 2006. 434-437.

VAUGHN JR, R. B. & MORRIS, T. Addressing Critical Industrial Control System Cyber Security Concerns via High Fidelity Simulation. Proceedings of the 11th Annual Cyber and Information Security Research Conference, 2016. ACM, 12.

VENABLE, J. The role of theory and theorising in design science research. Proceedings of the 1st International Conference on Design Science in Information Systems and Technology (DESRIST 2006), 2006. 1-18.

VON ALAN, R. H., MARCH, S. T., PARK, J. & RAM, S. 2004. Design science in information systems research. *MIS quarterly*, 28**,** 75-105.

WALLS, J. G., WIDMEYER, G. R. & EL SAWY, O. A. 1992. Building an information system design theory for vigilant EIS. *Information systems research*, 3**,** 36-59.

WEBER, S. Design Science Research: Paradigm or Approach? AMCIS, 2010. 214.

WEISS, M. & MOURATIDIS, H. Selecting security patterns that fulfill security requirements. International Requirements Engineering, 2008. RE'08. 16th IEEE, 2008. IEEE, 169-172.

WHITTEN, N. 1990. Managing software development projects. Formula for success. *New York: Wiley,| c1990*.

WIERINGA, R. Design science as nested problem solving. Proceedings of the 4th international conference on design science research in information systems and technology, 2009. ACM, 8.

WILSON, M. & HASH, J. 2003. Building an information technology security awareness and training program. *NIST Special publication*, 800**,** 50.

WOHLIN, C., HÖST, M. & HENNINGSSON, K. 2006. Empirical research methods in Web and software Engineering. *Web engineering***,** 409-430.

WOODS, D. R. 1996. Problem-based learning for large classes in chemical engineering. *New Directions for Teaching and Learning*, 1996**,** 91-99.

YANG, W. & ZHAO, Q. Cyber security issues of critical components for industrial control system. Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese, 2014. IEEE, 2698-2703.

YODER, J. & BARCALOW, J. 1998. Architectural patterns for enabling application security. *Urbana*, 51**,** 61801.

ZHU, B., JOSEPH, A. & SASTRY, S. A taxonomy of cyber attacks on SCADA systems. Internet of things (iThings/CPSCom), 2011 international

conference on and 4th international conference on cyber, physical and social computing, 2011. IEEE, 380-388.

ZINEDDINE, M. The dilemma of securing industrial control systems: UAE context. Information Technology for Organizations Development (IT4OD), 2016 International Conference on, 2016. IEEE, 1-6.