# Design and Evaluation of COFELET-based Approaches for Cyber Security Learning and Training

**Menelaos KATSANTONIS [1], Ioannis MAVRIDIS [1], Dimitris GRITZALIS [2*]**

[1] *Dept. of Applied Informatics, University of Macedonia, Thessaloniki, Greece*

[2] *Dept. of Informatics, Athens University of Economics & Business, Athens, Greece*

## Abstract

Cyber security education is becoming more important, as skilled cyber security professionals are highly sought. Though, cyber security education poses several weaknesses that limit the effectiveness of the delivered educational programs. In this light, game-based learning approaches provide a new prospect for cyber security education, as serious games have been utilized successfully in many fields (e.g., healthcare). However, cyber security game-based learning is a new field that lacks design standards and common methodologies. Towards this direction, the Conceptual Framework for eLearning and Training (COFELET) has been proposed, a framework that can be used as a guide for the design and evaluation of effective cyber security learning and training approaches. COFELET-based approaches (COFELET approaches) assimilate well known cyber security threat analysis and modeling standards as the means to create interesting educational experiences. To do so, the COFELET ontology that describes the key elements of COFELET's approaches, along with the appropriate classes and properties, is employed. To address high degree of complexity to design and evaluate, and to confront high demands in cost the presented work provides a proof of concept for the applicability of the COFELET framework and the feasibility of the design and evaluation process of COFELET compliant games (COFELET games); and it additionally provides the means for facilitating the creation of COFELET approaches. In particular, an extension of the COFELET ontology is proposed regarding the learning and the instructional aspects of a COFELET game; and the architectural and design aspects of a COFELET compliant game are presented including its structural elements, the major components and the life-cycle of a COFELET game. The paper also explores the design of HackLearn, a cyber security research prototype serious game designed for the application and review of the COFELET framework. The HackLearn's game design is put on the test of a preliminary evaluation scheme elaborated for the assessment of new cyber security game-based learning approaches and live competitions. The results show that the COFELET framework facilitates feasible and effective solutions and reveal the limitations of the HackLearn game.

*Keywords:* Cyber security, Serious Games, Design, Evaluation, Ontology, eLearning, Training, COFELET.

## 1 Introduction

Cyber security education faces many new and ongoing challenges in its effort to satisfy the required needs of the field (Katsantonis et al., 2019). Cyber security education challenges are primarily driven by the need for more cyber security personnel capable of facing the emerging threats and competing

[*] *Corresponding author.*
*email addresses: mkatsantonis@uom.edu.gr (Menelaos Katsantonis), mavridis@uom.edu.gr (Ioannis Mavridis), dgrit@aueb.gr (Dimitris Gritzalis)*

the cyber criminals in terms of knowledge and competencies. According to the International Information System Security Certification Consortium (ISC) (ISC, 2019) cyber security workforce needs to grow by 145% to meet the market demands. At the same time the cyber security incidents continually rise in numbers and fierceness (Risk Based Security, 2020), affecting the global economy and the national security (ENISA, 2019). In this light, game-based learning approaches provide a new anchor for cyber security education, as serious games have been proven effective educational tools already successfully applied in many fields (e.g., healthcare etc. (Wang, 2016)). However, as cyber security game-based learning and training is a new approach there are very few studies in the field (Hendrix et al., 2016) that lack design standards and common methodologies (Katsantonis et al., 2017b). For this reason, the Conceptual Framework for eLearning and Training (COFELET) framework has been proposed as a reference for developing effective cyber security learning and training approaches (Katsantonis et al., 2019).

COFELET is a multidisciplinary framework embracing several features for the creation of effective cyber security game-based approaches appropriate for the satisfaction of the requirements of the cyber security field. COFELET realizes cyber security education as an attractive and open subject for a broad spectrum of people, including young individuals and women. For this reason, it encompasses the game-based learning perspective, and it draws elements from live competitions (e.g., capture the flag or CtF competitions) and cyber security modelling techniques.

Moreover, COFELET envisages approaches that rely on sound learning theories and innovative teaching methods that advocate the effectiveness of COFELET compliant approaches (COFELET approaches). On this ground, COFELET complies with the activity theory to analyze and the interactions of the learners with the games; it assumes the layer learning approach (Katsantonis et al., 2019) (Greitzer et al., 2007) to apply cognitive principles and to enhance learning process; and it uses the continuous learning paradigm (Sessa and London, 2015) to engage learners in a cycle of learning, updating and reinforcing knowledge. Besides, COFELET assimilates well known cyber security models and methodologies (e.g., MITRE CAPEC's attack patterns (MITRE, 2020) and Cyber Kill Chain model or CKC (Lockheed Martin, 2020)), generally used in cyber security threat analysis and modeling, to form highly organized and parameterized learning and training environments.

To this end, it employs the COFELET ontology, an ontology that provides coherent descriptions of the elements that COFELET games embrace and their relationships (Katsantonis and Mavridis, 2019). COFEELET ontology describes the elements that model the actions attackers perform (i.e., the tasks) to unleash cyber security attacks and the tactics and techniques they employ to achieve their malicious objectives. In such way, COFELET games continually monitor the learners' efforts and they facilitate advanced assessment, scaffolding and adapting capabilities.

The above mentioned features of COFELET constitute a COFELET compliant game (COFELET game) a novel multidisciplinary approach promising to deliver effective cyber security learning and training. However, such multidisciplinary systems involve considerable degree of complexity for their design and evaluation and high costs of creation and maintenance. Under this prism, the research question of the presented study refers to the feasibility of the development of COFELET games and the manner that the design and the implementation of such games can be facilitated. In this context we present the design of the HackLearn game, a research prototype cyber security serious game designed for the application and review of the COFELET framework. In particular, we present the life cycle of a COFELET game indicating the major components such games embrace, the actors involved in the design and development of COFELET games and the elements that can be reused in the cyber security education domain.

The presented work also proposes an extension of the COFELET ontology, providing analytical descriptions of the additional elements required in COFELET approaches. A prototype scenario of HackLearn is presented exhibiting the manner that the presented elements can be utilized to provide COFELET compliant learning experiences. Finally, an evaluation scheme is utilized to carry out a preliminary evaluation on the HackLearn design in order to gain an appreciation on HackLearn's effectiveness, to prove the feasibility of its development process and to verify the applicability of the COFELET framework.

For the elaboration of the presented work, we adopted the design and creation research strategy (Oates, 2005). According to the design and creation strategy, we initially studied the manner that the COFELET ontology elements can be combined, stored, and reused. This process led us to the definition of new concepts bounded to the concepts of the COFELET ontology. Subsequently, we initiated the development of a prototype COFELET game for the evaluation of the COFELET framework in real world settings. We employed an iterative research process of analyzing, designing, implementing and testing the structural components of the prototype game (i.e., artefacts). During this process, we combined and generalized the produced artefacts and we created the architectural design of the prototype game, applicable for all COFELET games. To appreciate the effectiveness of the elaborated design we put into effect the analysis scheme that we presented in (Katsantonis et al., 2017a). At the end of this process, we came up with the presented artefacts recommended for the design and development of COFELET games.

The remainder of this paper is organized as follows: section 2 briefly provides the theoretical background of this work; section 3 presents the proposed extension of the COFELET ontology after a brief description of the COFELET ontology; section 4 presents the life-cycle of a COFELET game and a blueprint that facilitates the design and development of such game; section 5 states illustrative details on the design of HackLearn including a description of the prototype HackLearn scenario; section 6 presents the evaluation of the HackLearn design and the paper closes with the discussion and the conclusion sections.

## 2 Background

### *2.1 The COFELET Framework*

The COFELET framework (Fig. 1) specifies the main elements that have to be taken into consideration for the design and development of cyber security serious games, along with the interconnection of these elements in the games' structure. The primary concept of the COFELET framework is the task (represented in Fig. 1 by cycles). Tasks are the actions directed at the fulfilment of the game goals. Tasks are organized in Scenario Execution Flows (SEF) that describe the sequences in which tasks have to be performed and they are defined in analogy to attack patterns. Learners have to perform the proper sequence of tasks to successfully apply SEF according to the occurring conditions. Conditions represent prerequisites needed to perform tasks. During a game session, the game monitors and assesses the players' progress and it scaffolds the players' efforts through a hints system and a teaching contents provision system (scaffolding system).

At the end of a game session, the learner's performance is assessed and reviewed, and feedback is provided (i.e., achievements) to the learner. Then, the challenge of the game is tuned to the optimal level and the profile of the learner is updated. The subsequent scenario is selected according to learner's profile and history as well as the learning objectives, the educational environment and the learning strategy of the subsequent learning session.

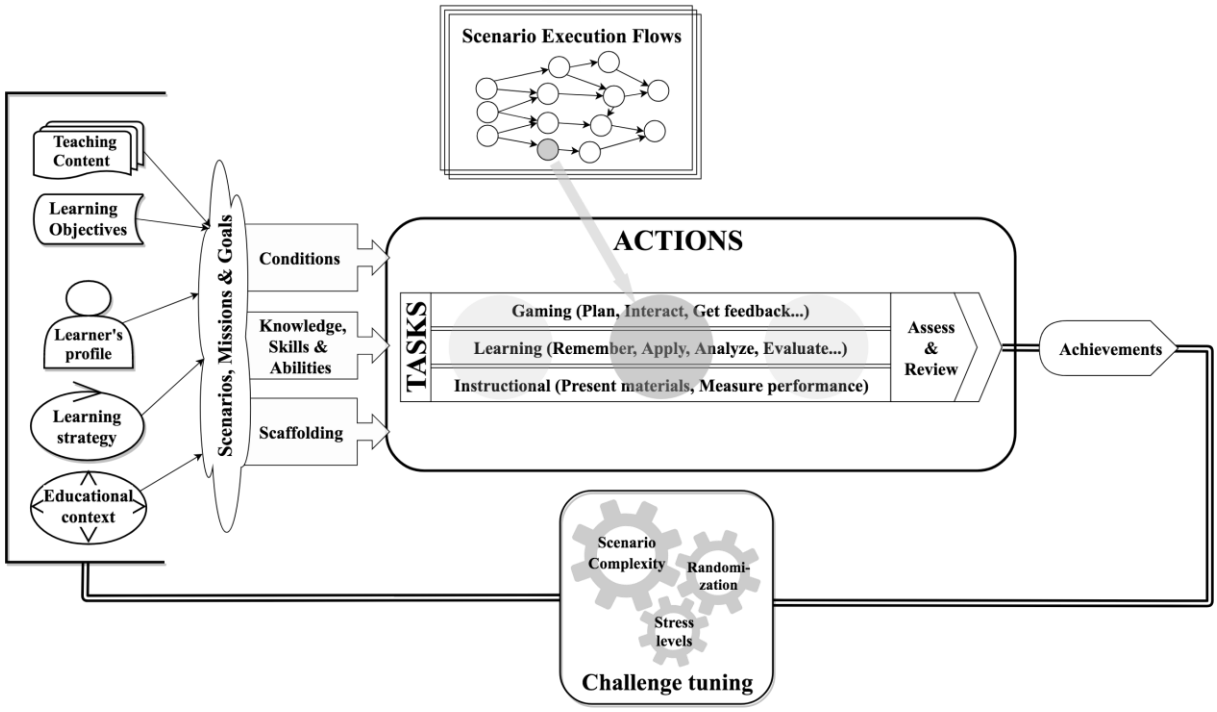


Fig. 1: The COFELET Framework (Katsantonis et al., 2019)

The COFELET framework complies with the Activity Theory Model for Serious Games (ATMSG) (Carvalho et al., 2015), an extension of the Learning Mechanic - Game Mechanic (LM-GM) model (Arnab et al., 2015), to facilitate the fusion of the learning aspect in serious games. The adoption of the ATMSG model in COFELET facilitates the systematic analysis, and organization of the games' components and the identification and classification of the actions and activities (i.e., a series of actions) that occur in the COFELET game. The identified activities are classified under the gaming, the learning and the instructional perspectives (the game perspectives) depicted in Fig. 1.

The gaming activities describe the learner's actions assuming the role of a gamer. For example, in a cyber security serious game such actions are the unleash of an attack and the acquirement of a flag. The learning activities refer to the actions a player performs assuming the role of a learner. Such actions in a cyber security serious game are the utilization of information (e.g., recall the sequence of actions and stages to unleash a cyber-attack), the utilization of cyber security tools, and the application of critical thinking to evaluate conditions and plan solutions (e.g., assess the applicability of tools and methods according to the game's context). Instructional activities refer to the actions carried out by the game aiming at providing scaffolding and feedback to support learners to achieve their learning objectives and reflect on their accomplishments. In particular, instructional actions refer to the providence of hints and the presentation of teaching contents related to the gaming and learning objectives of the game; the assessment of learner's efforts; and the presentation of achievements and scores or grades (i.e., feedback).

To ensure that the HackLearn game operates under the game perspectives, the game designers can initially oppose questions such as "what are the activities of the learner and what does these activities teach her?", "how does the game aid the learner in achieving the gaming goals and the learning objectives?", "how the monitoring and the assessment of learner's efforts is facilitated?". Subsequently, the game designers employ the ATMSG approach to analyze the activities, actions and components of the HackLearn game under the gaming, the learning and the instructional perspectives.

*2.2 Learning Strategies*

COFELET is established on the principles of the activity theory, through its conformity with the ATMSG model. Activity theory is a social constructivism theory used to analyze the components of interactive, composite and dynamic learning environments (e.g., COFELET games), as well as the learners' activities in such environments (Jonassen and Rohrer-Murphy, 1999). Besides, the learning process in a social constructivist learning environment is efficient as learners are encouraged to perform meaningful and realistic activities and to interact with the environment to solve problems (Vygotsky, 1978). The amount of passive activities such as reading, hearing and watching, is reduced and thus learners are not passive receivers of information as in traditional teaching methods (e.g., lectures, workshops, lab sessions) (Dewey, 1933) used in many cyber security education programs (Allen and Straub, 2015). Learners use prior knowledge to experiment with the system and make assumptions and errors that are not traumatic but pedagogically productive (Ausubel, 2000).

COFELET games utilize a layer learning approach to feature a good repertoire of learning strategies mapped to the Bloom's taxonomy levels (Fig. 2) (Katsantonis et al., 2019).

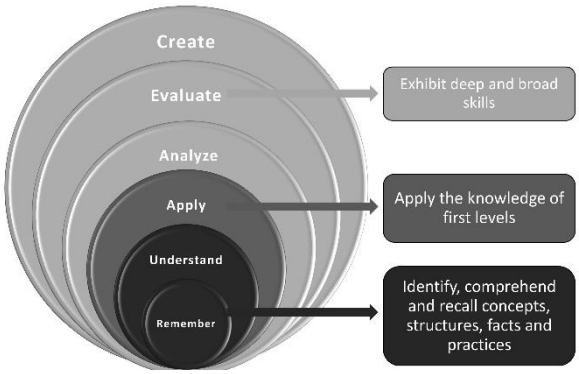


Fig. 2: COFELET layers mapped to the revised model of Bloom's taxonomy (Katsantonis et al., 2019)

In particular, the higher levels assume learning approaches that comply with modern learning theories and foster critical thinking, problem-solving abilities and analytical and creative skills. The higher levels reference learning sessions in which learners interact with realistic environments and try to solve genuine problems. On the contrary, lower and middle levels assume learning sessions in which learners perform simple activities such as comprehension and recall of concepts, utilization of tools and practice on tasks. Such approaches are important in cases that the objectives of the learning session include the update and reinforcement of critical cyber security knowledge and competencies. For example, for a cyber security professional working in an incident response team, it is important to immediately recall knowledge such as the port to protocol mappings or the utilization of cyber security tools to dump memory.

Nevertheless, COFELET adopts the continuous learning approach under two perspectives: a) learner has to try new experiences and challenges, b) learner has to try known things in new ways (Sessa and London, 2015). COFELET supports the first perspective by forming approaches in which the environment becomes increasingly immersive and complex and the learner has to confront new problematic cases tuned to her needs and cognitive level. On the other hand, under the viewpoint of the second perspective COFELET defines different contexts and conditions when the learner has to retry activities that update or reinforce knowledge and capabilities already possessed.

## 3 The Extended COFELET Ontology

### *3.1 The COFELET Ontology*

The design and development process of COFELET approaches is based on the COFELET ontology (Katsantonis and Mavridis, 2019). COFELET ontology provides the analytical descriptions of the key elements COFELET games need to include to interpret cyber security attacks in highly organized and parameterized learning environments. The key elements described in the COFELET Ontology are the Tasks, the Conditions, the Goals, the TaskNodes, the Scenario Execution Flows (SEFs) and the Scenarios. The Tasks, the Conditions and the Goals are the primary elements of the ontology (Fig. 3) denoting that an agent acts on an entity or an entity has a property. The primary elements are interpreted as quintuple statements of the form <subject entity, property, object entity or property value, source, destination> (e.g., the task <File transfer tool - sends - payload file, from learner's host to target host>).

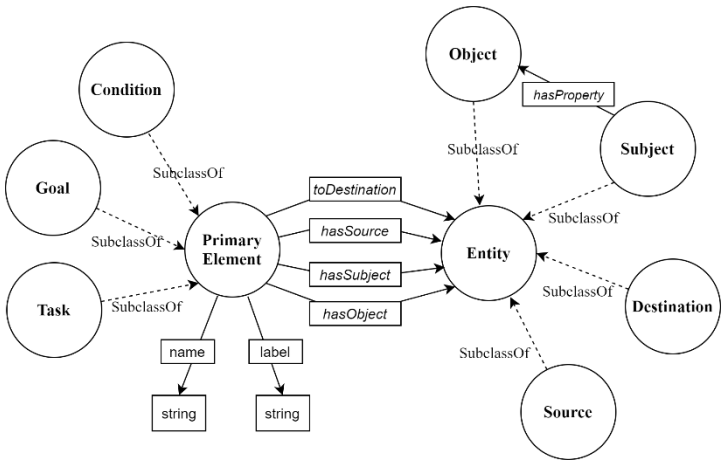


Fig. 3: COFELET Primary Elements (Katsantonis and Mavridis, 2019)

The primary elements are combined to form the SEFs and the COFELET scenarios. SEFs (Fig. 4) are created in analogy to the CAPEC's attack patterns as representations of cyber-attacks describing the sequence of tasks the learners follow, the conditions that have to occur in the game's context and the goal of the cyber-attack.

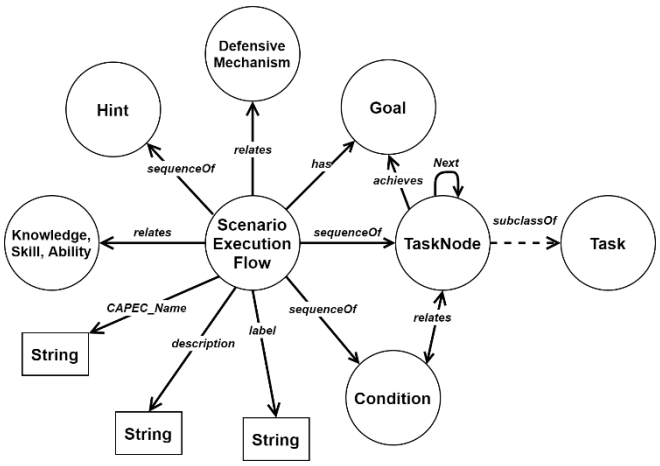


Fig. 4: COFELET Scenario Execution Flow (Katsantonis and Mavridis, 2019)

The COFELET ontology also describes the entities and the properties. Entities represent distinct concepts (e.g., agents, hosts, tools, commands) that lie in the context of the game, whereas properties symbolize relations between the entities. The COFELET ontology also defines the manner that the entities are organized in the context of COFELET games.

The presented work introduces an extension of the COFELET ontology including new elements that facilitate the learning and the instructional aspects of the game. In the remainder of this section, these new elements are described in detail along with their interconnections and their relationships with the initial elements of the COFELET ontology.

### *3.2 Roles*

A Role is a position or purpose that the learner has in an organization or an association. In COFELET games the developers organize the learning objectives (LO) and associate them with the workforce roles by utilizing the National Cybersecurity Workforce Framework (NCWF) (Newhouse et al., 2020) as a guide. In COFELET ontology role elements are represented by the Role class. The Role class contains the role description and a sequence of LO denoting the learning objectives that learners have to be able to achieve to successfully perform their duties in their jobs. The Role class associates the LOs it embraces with the following data properties:

- '*LO degree*': demonstrates the degree the learner possesses the knowledge or competency bound to the LO. The '*LO degree*' is a dynamic metric that changes overtime. In particular, the '*LO degree*' increases when the LO is exercised and decreases when it is not exercised for a period of time.
- '*durability*': indicates the time period after which the '*LO degree*' decreases.
- '*decay factor*': denotes how much of the '*LO degree*' is reduced after the '*durability*' period of time.
- '*last update*': is the date and time of the last '*LO degree*' update.

### *3.3 Learning Objectives*

The learning objective element is a brief statement describing the knowledge and the competencies the learner is expected to accomplish by the end of a learning process. The LO elements are created by the instructors in accordance to the knowledge, the skills, the abilities (KSA) and the tasks required by cyber security professionals to perform their duties in work and they are represented by the Learning Objective class (Fig. 5).

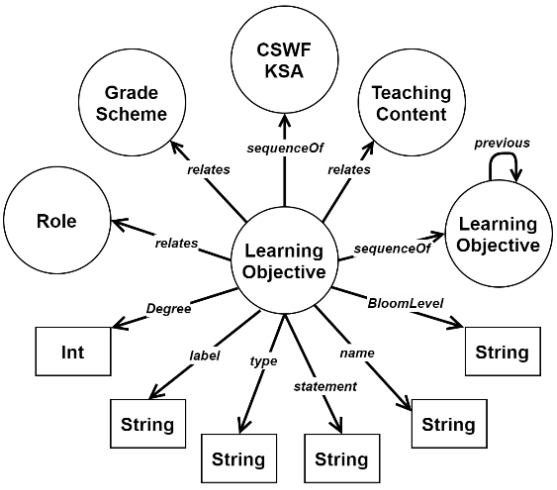


Fig. 5: COFELET Learning Objective

The LO class is a composite class containing several object and data properties listed in Table 1:

Table 1: The attributes of the LO class

| Attribute | Rational |
|---|---|
| Statement | denotes the statement of the LO in the form <Learner - property – object> |
| Type | indicates LO type that is knowledge, skill or ability |
| Name | the unique name of the LO |
| Label | the user-friendly name of the LO |
| Degree | associated with the '*LO degree*' data property defined in the Role class. It indicates how many points a successful LO achievement adds to the '*LO degree*' in the Role class. Instructors specify the value of the Degree in COFELET scenarios by taking into consideration the scenario's complexity and the mission's difficulty |
| Role | the associated role(s) based on the NCWF workforce role |
| BloomLevel | the mapping level in the Bloom's taxonomy |
| Learning Objectives | indicates a sequence of prerequisite LO |
| CSWF KSA | the KSAs in the NCWF that formed the basis of the LO |
| Teaching Content | the teaching contents associated with the LO |
| Grade Scheme | represents a rubric according to which the game assesses the learner's efforts |

### *3.4 Grade Scheme*

A Grade Scheme element specifies the grading scheme according to which COFELET games assess the learner's progress. The Grade Scheme element is associated with a LO element and it is represented by the GradeScheme class consisting of the following attributes:

- '*grade*': the points assigned to the learner after the assessment of her efforts. The grade is a fraction of the following attributes.
- '*assessed*': the number of assessments carried out on the associated LO.
- '*hints*': the number of hints provided to the learner with respect to the number of available hints in the scaffolding system.
- '*time*': the time period required to achieve the LO.
- '*actions*': the number of actions learner performed to achieve the LO.
- '*score*': the percentage applied to the '*Degree*' attribute of the LO class. The result determines how much the '*LO Degree*' attribute of the Role class will be affected.

In many cases the '*score*' and '*grade*' attributes have the same or proportional values. However, in some cases the instructor can assign a negative value to the '*score*' attribute in order to reflect a negative impact on the '*Degree*' attribute of the LO class as a disciplinary action when learners make the same mistake a number of times even after training and reinforcement (Nagarajan et al., 2012). On the contrary, the '*grade*' attribute does not take a negative value and it has range from 1 to 100.

### *3.5 Hints*

Hint elements represent the suggestions provided to the learners to help them achieve the game goals. A hint object has the following attributes:

- '*id*': unique id.
- '*name*': hint's unique name.
- '*label*': hint's user-friendly name.
- '*ord*': the order in which hints are displayed.

- '*time*': the period of time after which the learner is notified to take the hint or the hint is displayed automatically to the learner.

*3.6 Teaching Contents*

Teaching content elements are the materials (e.g., text, figures, and videos) aiding the learner to reinforce KSA or to assimilate the new knowledge and competencies. A content object includes the following attributes:

- '*id*': content's unique id.
- '*name*': content's unique name.
- '*label*': content's user-friendly name.
- '*text*': the text of the content.
- '*references*': references for additional information.
- '*lo_id*': association with the bounded LO.

*3.7 Scenarios*

Scenarios contain the appropriate information for the setup of a game session and they logically consist of three parts (Fig. 6). The first part contains the scenario's details such as the name, the label, the scenario's description and the difficulty level. The second part defines the cyberspace embracing a collection of entities and conditions (i.e., scenario's preconditions). The third part contains a sequence of steps corresponding to the stages of a multistage mission. Each step contains a subgoal, a set of conditions (e.g., pre- and post-conditions), a set of LO and a sequence of hints.

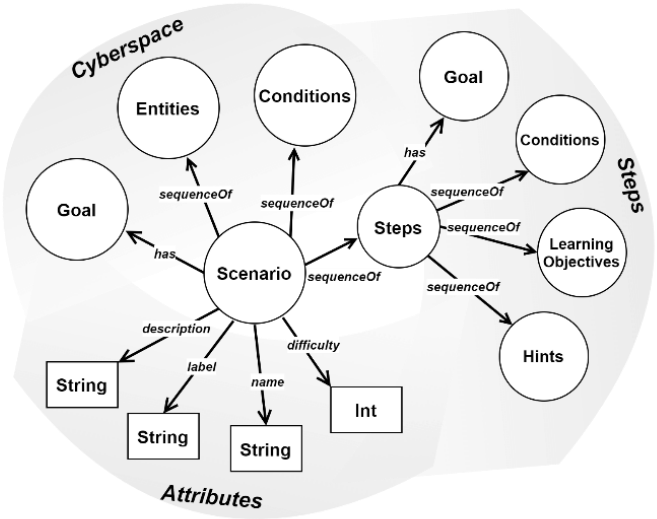


Fig. 6: COFELET Scenario

## 4 Design and Development of COFELET Games

In this section the architectural and the design aspects of a COFELET game are presented. A COFELET game has a modular architecture for two reasons: a) it contains individual components, which perform specific and clearly defined functions b) it contains structural elements (i.e., COFELET scenarios) formed by the combination of the elements described in the COFELET ontology (foundational elements). The foundational elements are the primary elements, the SEFs, the LOs, the teaching contents and the hints. In this light, the remainder of this section illustrates the life-cycle of

a COFELET game, presenting how the foundational elements, the structural elements and the game's components are organized in the structure of the game. Subsequently, the section presents the main actors involved in the development of a COFELET game along with the manner they have to cooperate to design and develop such games.

### *4.1 COFELET Game Life-cycle*

Fig. 7 illustrates the diagram of the life-cycle of a COFELET game (COFELET game life-cycle) including the elements and the components of such games and their organization in the structure of the game. The COFELET game life-cycle consists of the game's runtime phase and the build-time phase. The build-time phase contains the game foundations sub-phase, in which the foundational elements are created, and the game construction sub-phase, in which the structural elements (i.e., COFELET scenarios) are formed. In the runtime phase of a COFELET game life-cycle, the major components of the game are depicted along with the functions they perform and their interconnections.

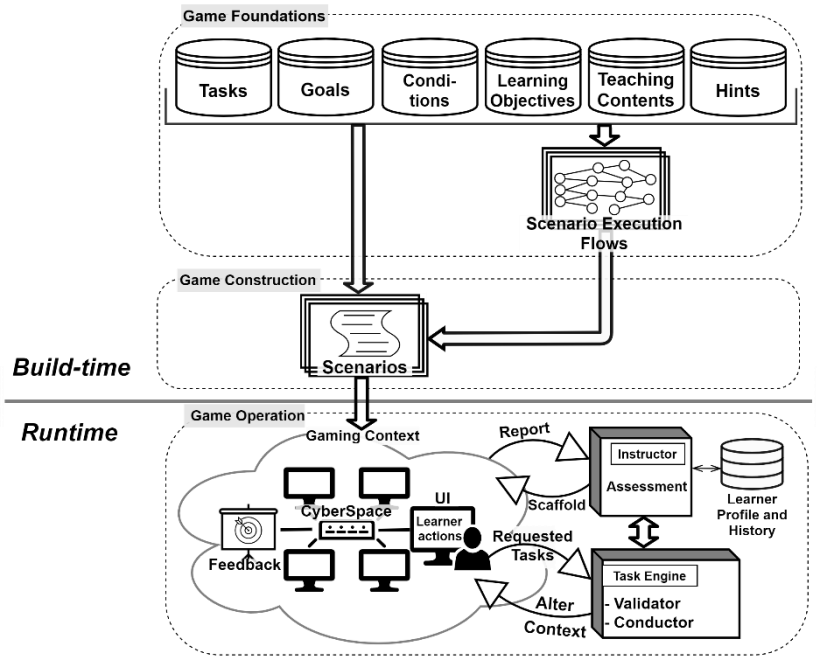


Fig. 7: The life-cycle of a COFELET game

### *4.2 Actors*

Fig. 8 depicts the use case diagram of the actors involved in the life-cycle of a COFELET game. The main actors involved in the development of a COFELET game are the game developers, the cyber security specialists and the instructors. Game developers create the games by implementing the designs of the cyber security specialists and the instructors. Cyber security specialists are cyber security domain experts having deep knowledge of cyber security methodologies and models (e.g., the CKC model).

Cyber security specialists utilize the COFELET ontology to design the primary elements, the attack patterns and the strategies that will be interpreted in the games. Instructors are educators aware of the NCWF and the roles, the tasks and the KSAs it defines. Instructors complement the work of cyber security specialists by adding the elements that determine the learning and the instructional perspectives of COFELET games (i.e., the LOs, the hints and the teaching contents).

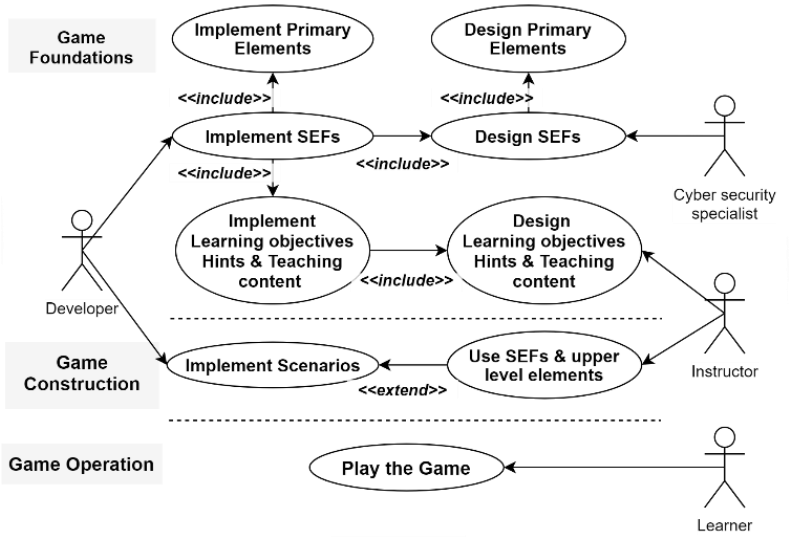

Fig. 8: Use case of the actors involved in the life-cycle of a COFELET game

*4.3 Build-time*

During the '*Game Foundations*' of the '*Build-time*' phase, game developers cooperate with the cyber security specialists and the instructors to create the foundational element repositories depicted in Fig. 7. Foundational elements are stored in a manner that facilitates their adoption in various games and educational contexts.

During the '*Game Construction*' of the '*Build-time*' phase, instructors create the COFELET scenarios. COFELET scenarios are created by utilizing the foundational elements stored in the COFELET repositories. COFELET scenarios describe in-game entities by providing the necessary properties for imitating the behavior of real devices. Some entities have attributes with randomized values that change from session to session. COFELET scenarios also contain additional elements when instructors need to add extra functionalities and features. In such cases, instructors need to cooperate with game developers during the '*Game Construction*'. For example, HackLearn's prototype scenarios define the 'Question' elements representing the questions issued during the game play. HackLearn's questions regard employed attack patterns, cyber security concepts and they explicitly are associated with the LO. Apart from the additional elements, instructors cooperate with the game developers in the game's construction phase to combine the foundational elements and define new SEF for AP that are not present in the SEF repository.

*4.4 Runtime*

The 'Runtime' phase of the COFELET game life-cycle depicts the major components of a COFELET game that are the gaming context, the task engine and the instructor component. The gaming context contains the user interface façade (UI) and the game's cyberspace. The cyberspace is the virtual environment in which learners perform their actions and they unleash their cyber-attacks. The cyberspace embraces numerous game entities such as the learner's host, networks, target hosts, servers and services, firewalls, files etc. The UI depends on the genre of the COFELET game. For example, the UI of a hacking simulation game (e.g., the HackLearn) usually includes a command terminal in which the learner enters commands along with a set of windows that embrace additional functionalities (e.g., display information, send messages etc.). On the other hand, the UI of a card game includes a card deck and a game menu. The cyberspace provides feedback to the learner through the facilities that it embraces (e.g., the terminal in the learner's host) and through the game's UI.

The Task Engine is a task operator that conducts the performed tasks and provides feedback to the learner through the cyberspace. The Task Engine consists of a task validator and a task conductor. The task validator confirms that a task belongs in the sequence of tasks of the employed SEF and validates that a task is executable by inspecting the occurring conditions. In multi-step scenarios, the Task Engine checks whether a task initiates a new SEF and indicates the enrollment of a new SEF. The task conductor virtually executes a task and checks whether the execution of a task provokes the fulfillment of a goal or a mission. The task conductor also sets the post conditions of the executed task and communicates with the gaming context and the instructor component. The instructor component assesses the learning session and scaffolds the learner's efforts. To do so, it monitors the learner's progress and it acquires the necessary information from the Task Engine and the gaming context. The details that the instructor component acquires include the learner's actions, the tasks performed, the goals achieved and the current SEF that the learner applies.

Moreover, the instructor component manages the appropriate foundational elements such as the scenario's hints, the teaching contents and the LO that the learner has to achieve. Finally, instructor component has access to the game's back end storage facility (e.g., a database, or a collection of XML files) and queries information regarding the learner's profile and the learning and training history. During the game play, the instructor component scaffolds the learner's efforts through the provision of hints and teaching contents that are associated with the LO learners have to achieve. For example, in HackLearn the instructor component counts the time and monitors the learner's progress. Whenever, the time period is beyond a time threshold specified in the game's scenario by the instructor (i.e., in the '*time*' attribute of a hint object), instructor component notifies the learner and provides the appropriate hint(s). On the other hand, when the learner achieves a goal the instructor component assesses the learner's fulfillment by applying a grading scheme specified by the instructor in the COFELET scenario (i.e., the GradeScheme objects). Subsequently, it stores the assessment details in the back-end storage facility and it updates the learner's profile.

The runtime phase of the COFELET game life-cycle exhibits the manner according to which COFELET games realize the game perspectives. Particularly, a COFELET game renders the learner actions in two places: in the cyberspace and in the Task Engine. The cyberspace imitates the real world and it interprets the learner actions under the gaming perspective. For example, the cyberspace of the HackLearn game imitates the settings of a live competition by embracing the suitable entities with the appropriate functionalities and attributes. In such contexts, the learners assume the role of a live competition's participant. The learners' actions are additionally interpreted under the learning perspective as the requested tasks are passed to the Task Engine. In the Task Engine the requested tasks are compared with the SEF's tasks that are explicitly related with the learning and the instructional aspects of the game (e.g., the LO, the hints and the teaching materials). In such way, the game translates learner's actions to accomplishments of LO related to the acquirement of KSA. Nevertheless, the Instructor component assumes the role of an instructor by carrying out activities that take place in the game under the instructional perspective. Such activities are related with the assessment of learners' efforts and the provision of hints and teaching contents explicitly related with the learners' tasks.

## 5 Excerpt of the HackLearn Design

In this section an excerpt of the HackLearn design is presented along with a set of instances of the COFELET ontology objects presented in section '*3. The Extended COFELET Ontology*'. Initially, the genre of HackLearn is described and subsequently a prototype HackLearn scenario is presented providing an insight of the way the game operates.

### *5.1 HackLearn Genre*

Hacking simulation games or hacking simulators have been around for many years and they are becoming more popular over the past years. At the moment, the Steam game distribution platform (Valve, 2020a) offers more than 20 commercial entertainment hacking simulation games such as 'HackNet', 'hack_me' and 'NITE Team 4'. Hacknet (Fellow Traveller, 2020) is one of the most popular hacking simulators with 1-2M owners, more than 70K followers and more than 10K positive comments (Valve, 2020b).

HackLearn includes several features typically found in hacking simulation games such as a Unix-like terminal in which players type and execute text-based commands; simulations of cyber security attacks; representations of common cyber security entities and concepts (e.g., hosts, firewalls, services); role playing experiences as a player assumes the role of a hacker that faces various challenges. HackLearn draws many elements from CtF competitions as the learners unleash cyber security attacks during the game-play, exercise their knowledge and skills, collect flags and points, and try to beat the clock. Nevertheless, HackLearn is a hacking simulator game that embraces advanced features (e.g., adaptability, dynamic assessment, scaffolding, conformance with sound theories) that have not yet been included in CtF competitions and in cyber security game-based approaches.

### *5.2 Prototype Scenario*

The elaborated prototype scenario is a multi-step complex scenario with target group computer scientists with a strong background in cyber security aiming at acquiring knowledge and competencies of the 'Vulnerability Assessment Analyst' role of the NCWF. The scenario has a narration in the description attribute of the scenario object that describes the mission, provides a story that supports the mission and offers clues that will help the learner to achieve the scenario's goals. The LO defined in the prototype scenario are listed in Table 2 (for brevity only the statement attributes of the LO objects are listed) along with the matching Bloom level and a LO code. Table 3 provides a list of the scenario's steps with the corresponding LO, whereas Fig. 9 provides a diagram of the scenario's cyberspace. Most of the LO presented in Table 2 belong to the application level of the Bloom taxonomy, as the scenario mainly exercises skills in penetration testing.

However, the L3 of step S3, the L6 of step S4 and the L12 of step S8 belong to the higher COFELET layers (fig. 2) because they require deep knowledge and competencies. Specifically, in step S3 learner has to appraise which host and service is most likely to have a vulnerability, otherwise she will end up searching all the services in the exploit-db. In step S4 learner has to improvise to exploit the password recovery mechanism after the realization that the guest account is inactive; and in step S8 learner creates a plan of an APT attack (creation level) by applying the CKC model (application level). On the contrary, the L13 LO belongs to the low COFELET layers, as the learner distinguishes the port states. The L13 LO is associated with the question '*Is a filtered target port considered opened or closed?*' prompted in S2 step. The prototype scenario contains several low level LOs associated with questions prompted during the game-play that they are not listed in Table 2 for brevity.

Table 2: The learning objectives of the prototype COFELET scenario

| Code | Bloom Level | LO statement |
|---|---|---|
| L1 | Application | Learner utilizes network analysis tools (i.e., host scanners) to identify alive hosts in a network |
| L2 | Application | Learner utilizes network analysis tools (i.e., port scanners) to determine the status of ports on the targets |
| L3 | Evaluation | Learner identifies potential points of vulnerability |
| L4 | Application | Learner finds exploits in the exploit-db |
| L5 | Application | Learner utilizes remote access tool to connect to a remote target (via a shell) |

| | | |
|---|---|---|
| L6 | Analysis, Application | Learner exploits password recovery mechanism |
| L7 | Application | Learner uses an exploit to abuse authentication and escalate her privileges |
| L8 | Application | Learner creates a weapon file |
| L9 | Application | Learner utilizes a file transfer tool with privileged rights to deliver a weapon file to the target |
| L10 | Application | Learner utilizes a reverse shell |
| L11 | Application | Learner finds the flag file |
| L12 | Creation, Application | Learner applies the stages of the CKC model |
| L13 | Comprehension (Understand) | Learner distinguishes the port states |

Table 3: The steps of the prototype HackLearn scenario

| Code | Label | Rational | LOs |
|---|---|---|---|
| S1 | Host discovery | Learner performs a host discovery attack pattern to find potential target hosts. | L1 |
| S2 | Port scan | Learner scans the ports of the hosts discovered in step S1 to find information of the services running on these hosts. Among the discovered services, the learner finds the target service. | L2, L13 |
| S3 | Exploit search | Learner searches the exploit database to find an exploit that can be used on the vulnerable service discovered in step S2. However, the exploit requires that the attacker has credentials of a legitimate user (e.g., guest). | L3, L4 |
| S4 | Password recovery exploitation | Learner uses a remote connection tool to connect to the service and finds out that the guest account is inactive. Subsequently, learner finds a weakness in the password recovery mechanism and exploits it to get the password hint of a legitimate user. Then, learner excavates user's personal information to find the user's credentials. | L5, L6 |
| S5 | Authentication mechanism abuse | After the realization that the acquired credentials refer to a user with low privileges, learner utilizes the credentials with the exploit found in step S3 to connect to the target host. Then, learner inspects the files and the directories of the target and finds out that she does not have access to the directories and files of the system, and thus she cannot find the flag. However, learner has privileged rights on a distinct directory containing an exe file. | L7 |
| S6 | Weapon creation | Learner utilizes a payload maker tool and the exe file discovered in the previous step as a template to create a weapon file | L8 |
| S7 | Weapon delivery | Learner connects to the target and delivers the weapon | L9 |
| S8 | Backdoor utilization | Learner utilizes the backdoor, connects to the host with administrator rights and discovers the flag. The mission is achieved | L11, L12, L13 |

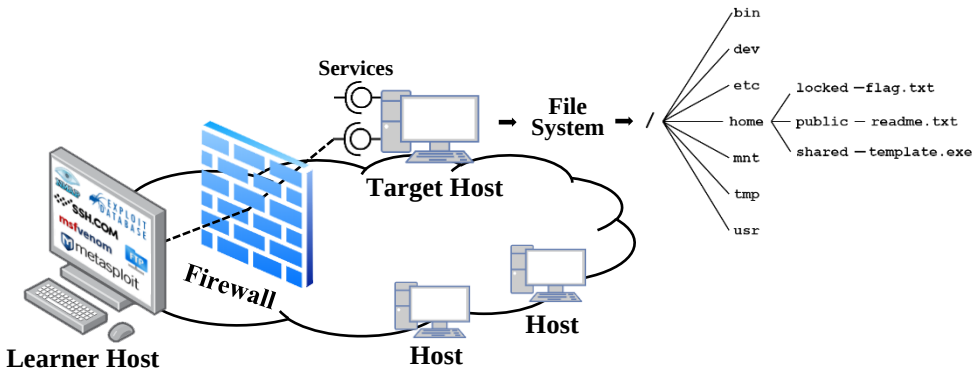


Fig. 9: Cyberspace of the prototype scenario

Table 4 states in more detail the attributes and the rational of the L2 LO associated with the S2 step, as well as the hints associated with this step.

Table 4: The attributes of the L2 learning objective '*Learner utilizes network analysis tools (i.e., port scanners) to determine the status of ports on targets*'.

| Attribute | Value and Rational |
|---|---|
| Type | Skill |
| Name | PortScanLO_01 |
| Label | Port Scan Learning Objective |
| Degree | 34 |
| Role | '*Penetration tester*', '*Vulnerability Assessment Analyst*' and '*Target Network Analyst*'. '*Penetration tester*' role is not included in the NCWF. However, it combines knowledge and competencies of the '*Vulnerability Assessment Analyst*' and '*Target Network Analyst*' of NCWF |
| BloomLevel | Application |
| Learning Objectives | L1, as the learner has to know how to discover a host before she scans its ports |
| CSWF KSA | It is based on the S0081 skill of NCWF '*Skill in the use of penetration testing tools and techniques*' and the S0051 skill of NCWF '*Skill in using network analysis tools to identify vulnerabilities. (e.g., fuzzing, nmap, etc.)*' |
| Teaching Content | The material is adopted from the '*Port Scanning*' attack pattern of CAPEC (https://capec.mitre.org/data/definitions/300.html) |
| Grade Scheme | The Grade scheme is an array of GradeScheme objects as the objects described below:<br>GradeScheme1. "*times assessed='0-1', hints='0', time='1-50', actions='0-3', score='100', grade='100'* ".<br>GradeScheme2. "*times assessed='4-6', hints='1-4', time='1-50', actions='0-3', score='-10', grade='1'* "<br>The GradeScheme1 object denotes that the first or second time that the LO is achieved the learner will have grade 100%, if no hints are taken, if the time taken from the last goal is 1 to 50 in seconds and if the learner performs 0 to 3 actions (excluding the task that exercised the LO). The score 100 denotes that the achievement of the LO adds 34 points (i.e., '*Degree*' attribute of LO object) to the corresponding skill in the profile of the learner (i.e., '*LO Degree*' attribute of role object).<br>The GradeScheme2 object denotes that the learner will have a penalty score of 10 if the 4th to 6th time that the LO is assessed, she will acquire hint(s). |
| Hints | Hint 1: Text = 'Identify the open ports of the targets', time='*120*' seconds<br>Hint 2: 'Use a port scanner tool that sends probes to an IP address/port and determines the status of the port', time='*120*'seconds<br>Hint 3: 'Scan your target(s) using a port scanner (e.g., nmap) and the appropriate scan option (e.g., TCP Scan option)' , time='*120*' seconds<br>Hint 4: 'In the terminal issue the command: *nmap –sS \|target\|*' , time='*120*'seconds |

The scenario's cyberspace is a collection of entity objects with the appropriate attributes and functionality to imitate the behavior of the real devices. Fig. 10 depicts a UML class diagram for the entities of the scenario's cyberspace.

### *5.3 Command Execution*

HackLearn includes a terminal in which players enter text-based commands that utilize Unix-like tools to perform the game's tasks (the command execution action). The sequence diagram in Fig. 11 shows the manner that the components depicted in the runtime part of Fig. 7 (in section "*4.1 COFELET Game Life-Cycle*") interact to perform command execution actions.

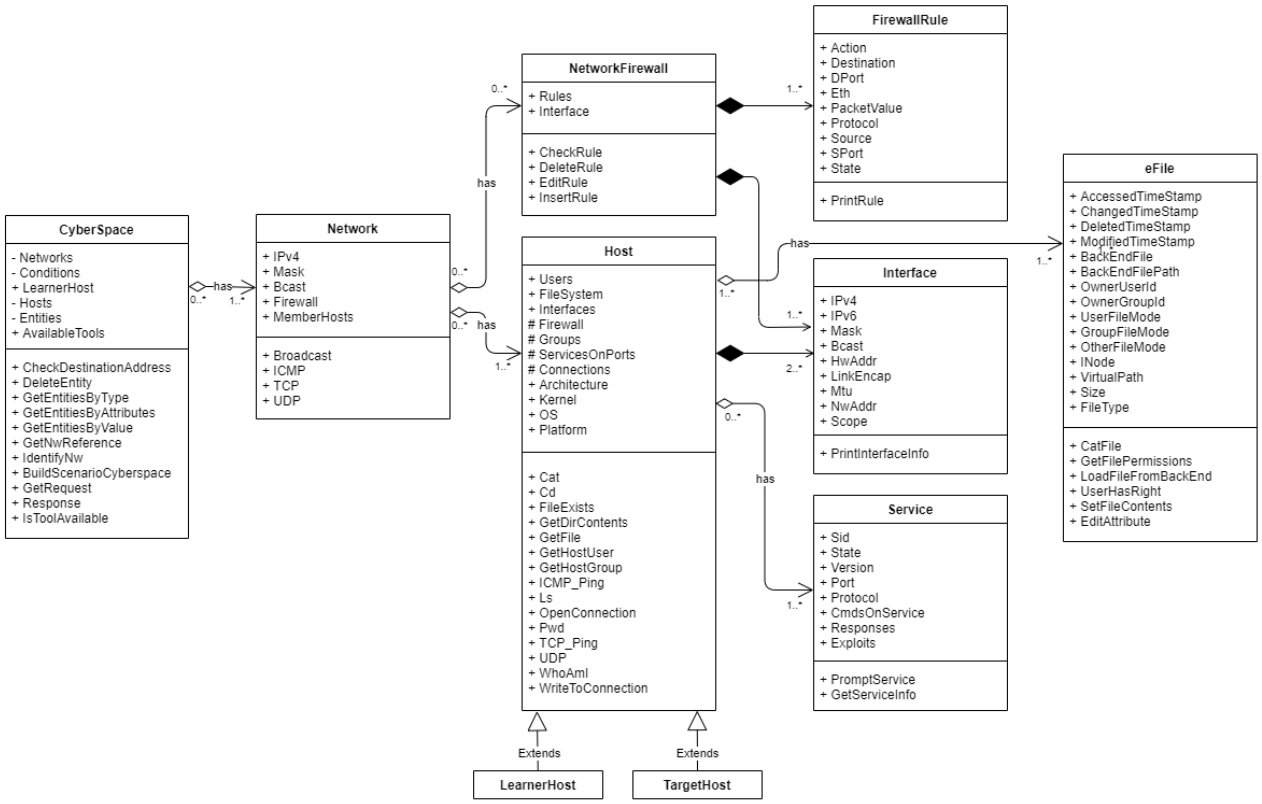


Fig. 10: UML Class diagram of the prototype scenario's entities

Once the learner enters a command in the UI, the terminal renders the command's arguments and passes the command to the appropriate tool that is built in the terminal. The tool makes all the appropriate audits, reports the learner's action to the instructor component and passes the corresponding task to the Task Engine. Then, the tool gets the response and performs the command in the cyber space that alters the gaming context. Finally, the learner receives feedback from the UI (e.g., scores, visualizations, sounds etc.) and from the terminal in textual form.

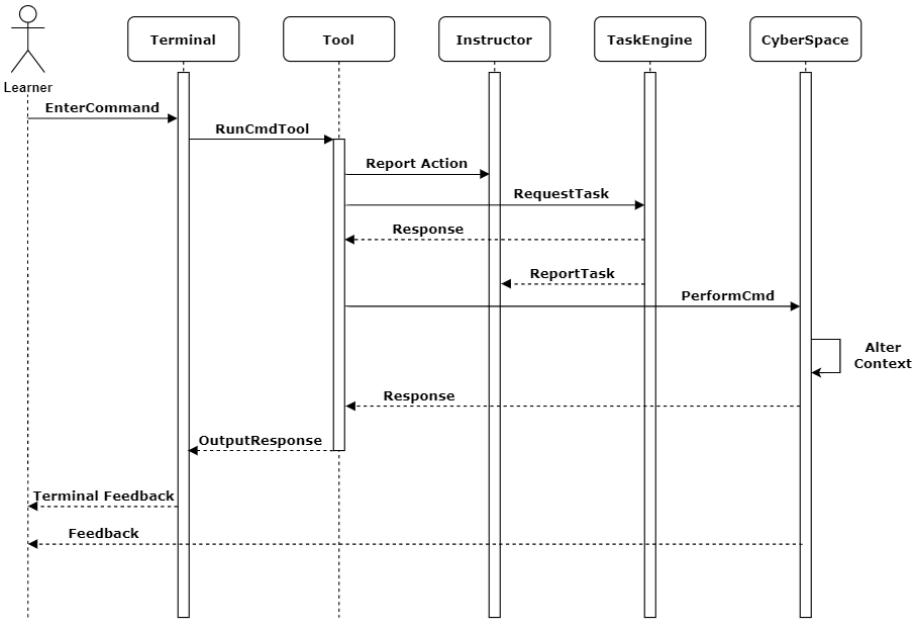


Fig. 11: Command execution sequence diagram

## 6 Evaluation

### *6.1 The Evaluation scheme*

The evaluation scheme of the presented study is based on the analysis scheme presented in (Katsantonis et al., 2017a), a scheme for conducting preliminary evaluations on new cyber security live competition approaches.

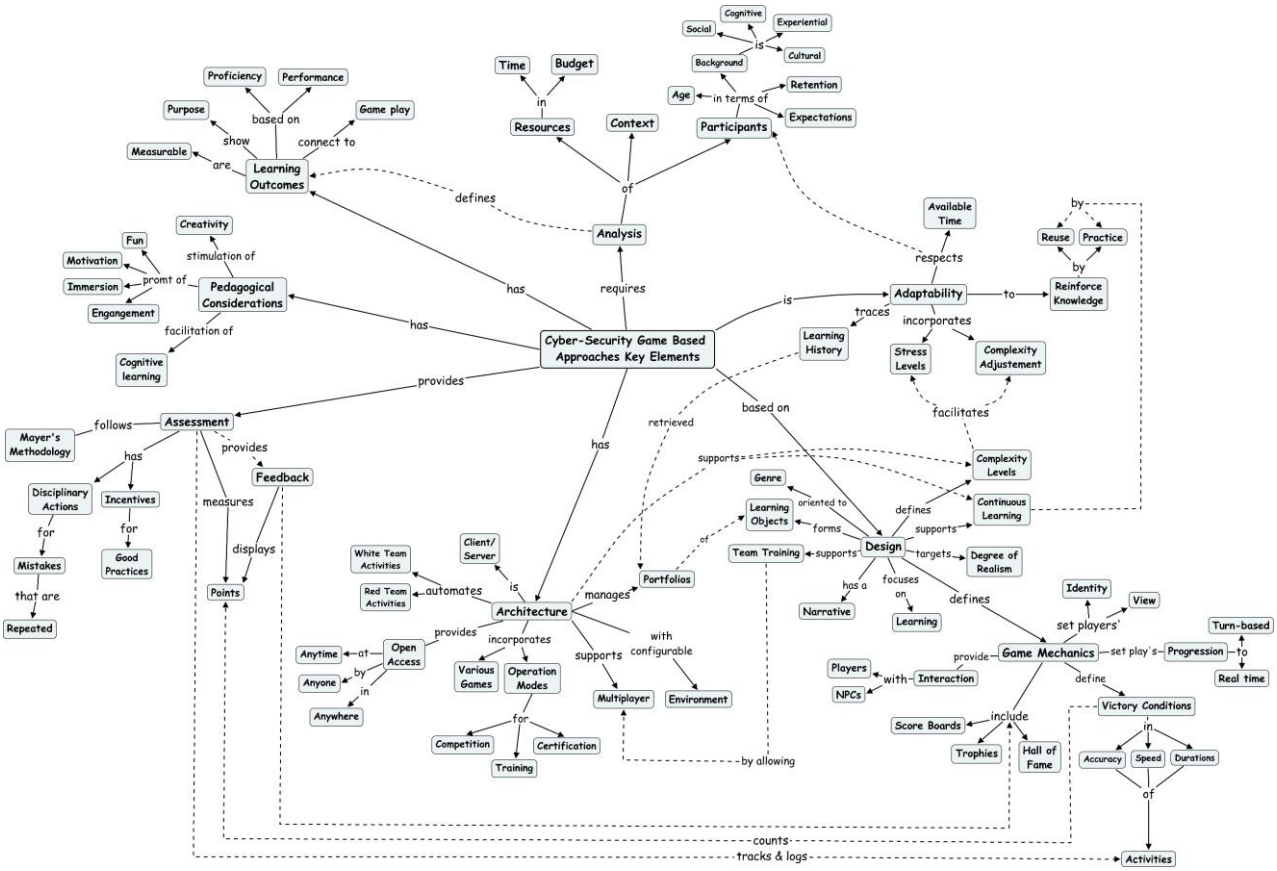


Fig. 12. Concept Map of Cyber Security Game Based Approaches Key Elements (Katsantonis et al., 2017b)

The scheme presented in (Katsantonis et al., 2017a) employs a concept map and a categorization of challenges as an assessment tool for the deduction of assumptions regarding the feasibility and the educational impact of new live competitions approaches, as well as the effectiveness of these approaches in coping with the identified challenges.

The evaluation scheme utilized in the presented work is based on the concept map of cyber security game-based approaches key elements (GBL concept map) depicted in Fig. 12 (Katsantonis et al., 2017b). Besides, as the HackLearn game draws many elements from live competitions domain, the utilized evaluation scheme also utilizes the concept map of live competitions' technological and pedagogical characteristics (CtF concept map) depicted in Fig. 13. The utilized evaluation scheme also employs the identified problems and issues of the field presented in (Katsantonis et al., 2017a) and in (Katsantonis et al., 2019).

The GBL concept map has the main role in the evaluation as it captures the characteristics of current cyber security game-based approaches derived from the current studies and frameworks. The CtF concept map has a secondary role, as only the '*Pedagogical Benefits*' and the '*Assessment*' segments are utilized because they are consistent with the COFELET approach.

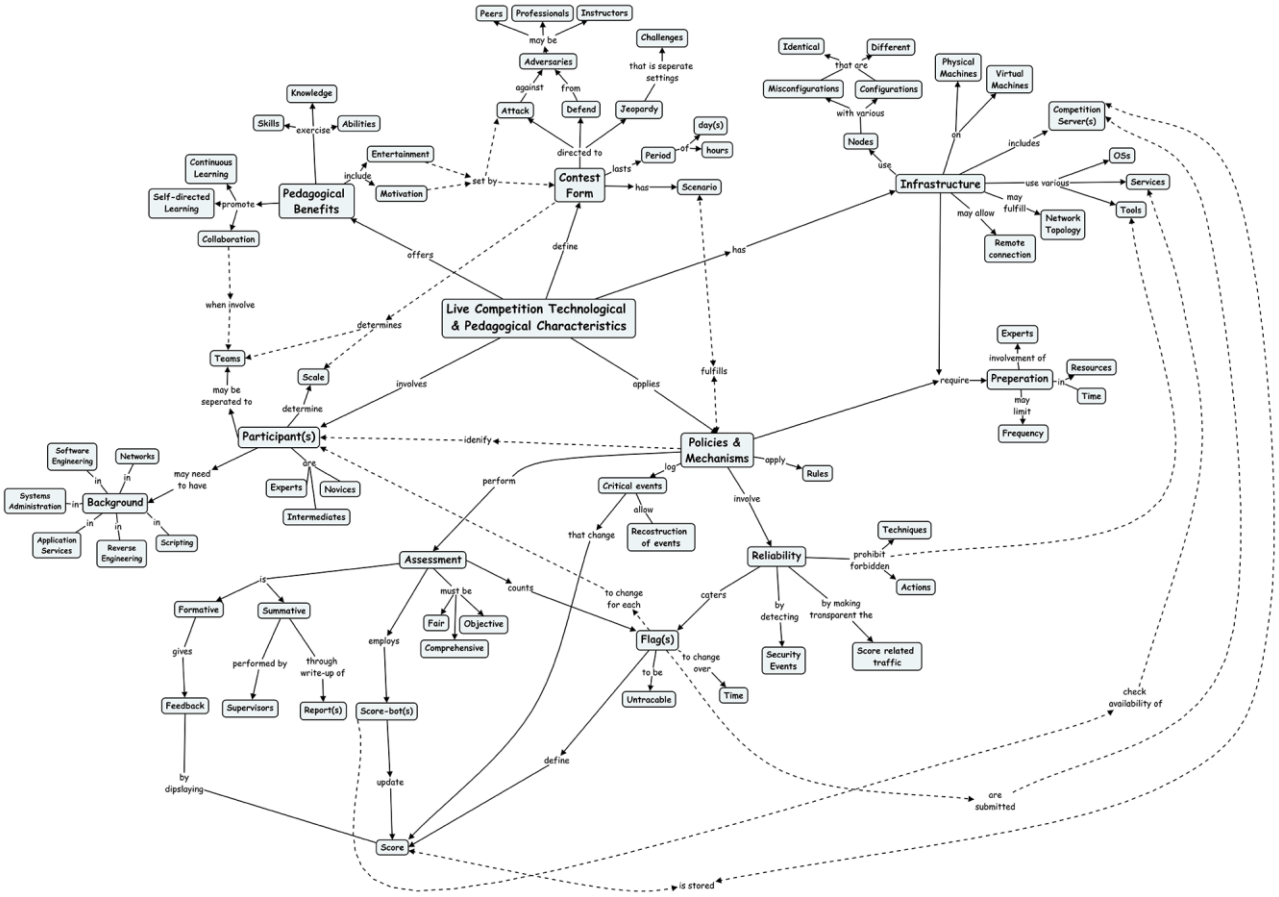


Fig. 13: Concept Map of Live Competitions Technological and Pedagogical Characteristics (Katsantonis et al., 2017a)

In the remainder of this section, the design of the HackLearn game is put on the test of the evaluation scheme stated above. In particular, the HackLearn game is resolved into its elements, and its pedagogical effectiveness is appreciated by comparing the characteristics it contains with the characteristics of the concept maps and in accordance to the problems that it tries to solve. The results of the evaluation are presented in Table 5. Table 5 consists of seven parts, six parts in analogy to six segments of the GBL concept map and one part for the issues and challenges of the filed.

Table 5 references 6 out of the 8 segments of the GBL concept map, as the '*Analysis*' segment is combined with the '*Adaptability*' segment and the '*Game mechanics*' segment is considered part of the '*Design*' segment. The column '*Characteristics*' contains the characteristics of the GBL concept map and the characteristics of the CtFs concept map. For brevity, the sibling characteristics (e.g., formative and summative assessment) are presented in the same rows of Table 5.

The characteristics and the segments listed in Table 5 have subscript specifications indicating the concept map that they are adopted from (i.e., '*GBL*' for the characteristics adopted from the GBL concept map and '*CTF*' for the characteristics adopted from CtFs concept map). The column '*Support*' indicates whether the characteristic is supported (symbol '✓'), not supported (symbol '✗') or merely supported ('?'), whereas the column '*Rational*' explains the rational of the '*Support*' specification.

Table 5: HackLearn's evaluation

| Characteristics | Support | Rational |
|---|---|---|
| **Segment 1: Pedagogical Considerations GBL and Pedagogical Benefits CTF** | | |
| Cognitive learning GBL | ✔ | HackLearn is based on the cognitive learning theories, as it constitutes an educational environment where learners are able to perform actions, experiment, reflect on their deeds, utilize new practices and assimilate new KSAs. Moreover, HackLearn fosters critical thinking and problem-solving capabilities, as the learner appraises the context of the game plans and executes an attack based on the CKC model. |
| Creativity GBL | ✔ | In HackLearn, instructors define scenarios in which learners have to think outside of the box and exercise new skills. For example, in the step S5 of the prototype scenario the learner has to apply a genuine attack pattern in order to analyze the manner the password recovery mechanism of the target service operates, retrieve the password hint, excavate user's personal information and get the credentials required to proceed to the next step. |
| Engagement, immersion, motivation and fun GBL | ? | HackLearn adopts the attack concept of live competitions, an important factor that enhances the motivation and the entertainment factors (Chung and Cohen, 2014). Additionally, it draws elements from role playing games that reinforce the engagement and immersion characteristics, as learners assume in-game roles and maintain profiles containing collections of KSAs. Unlike live competitions, the fun and motivation factors are affected by the employed instructional learning approach, as learners are obliged to follow the game's scenario elaborated by the instructor. |
| Continuous learning GBL | ✔ | HackLearn implements a continuous learning approach, as the game is 'always-on' providing the means for the organization of periodical learning sessions. In learning sessions, learners acquire new KSAs or they exercise the KSAs they already possess (adopted KSAs). To regularly exercise the adopted KSAs, an instructor can implement a policy of decreasing the '*LO Degree*' values in the learner's profile for LOs that have not been achieved for a specified period of time (specified in the 'last update' attribute of Role objects). Consequently, the learner has to periodically repeat training sessions that exercise KSAs bound to LOs with low '*LO Degree*' values, and thus she enters in a continuous lifecycle of learning, updating and reinforcing KSAs. HackLearn provides the opportunities for learners to exercise their adopted KSAs in new ways (Sessa and London, 2015) by altering the narratives, the cyberspaces and the conditions of the sessions and by utilizing randomization in the attributes of the entities (e.g., network's IP address). |
| Self-directed learning CTF | ✘ | On the contrary with the live competitions which promote self-directed learning, HackLearn promotes instructional learning. |
| Exercise of knowledge, skills and abilities CTF | ✔ | In HackLearn learners exercise techniques and basic skills such as the discovery of live hosts in a network (in steps S1), the scan of the target's ports (in step S2) and the creation of a weapon payload file (in step S6). |
| Collaboration CTF | ✘ | HackLearn is a single player game and lacks the promotion of collaboration among learners in the context of the game. |
| **Segment 2: Learning Outcomes GBL** | | |
| Connection to the game-play | ✔ | HackLearn infuses the LOs in the game-play and associates the gaming goals with the learning objectives (analysis presented in sub-section '*4.4 Runtime*'). |
| Learning outcomes show purpose and they are measurable | ✔ | HackLearn's LOs are based on the NCWF's KSAs, they are measurable and they have clear purpose. |
| Assess proficiency and performance | ✔ | The assessment of the LOs is based on the measurement of the learners' performance as it involves the recording of the tasks' details (e.g., duration, number of repetitions) associated with the LOs. Moreover, the HackLearn game aims at assessing the LOs in various in-game contexts to ensure the proficiency in exercising the cyber security knowledge and skills under different conditions. |
| **Segment 3: Architecture GBL** | | |
| Open access | ✔ | HackLearn provides open access as anyone can use it anytime from anywhere. |
| Configurable environment | ✔ | HackLearn allows the full configuration of the environment in which the learner operates, mainly through the specification of the cyberspace and the conditions in the COFELET scenarios. |
| Manage portfolios of learning objects | ✔ | HackLearn's repositories can be considered portfolios of cyber security learning objects which can be adopted in various learning and training environments. |
| Multiplayer | ✘ | HackLearn only operates in single player mode. |

| | | |
|---|---|---|
| Modes of operation | ? | HackLearn operates in training mode, but it does not support certification and competition mode. |
| Incorporation of various games | ✗ | HackLearn is not a game suite and it does not incorporate a collection of different genre games with different user interfaces and characteristics. |
| Automation of red team and white team activities | ✔ | HackLearn requires learners to perform red team activities and it can automate white team activities. |
| **Segment 4: Design and game mechanics** GBL | | |
| Orientation to genre | ✔ | HackLearn is a hacking simulation game (justified in the '*5.1 HackLearn genre*' sub-section). |
| Team training | ✗ | HackLearn does not support team training |
| Focus on learning | | HackLearn complies with the ATMSG model that facilitates the assimilation of the learning aspect in the game's design. |
| Realism | ? | HackLearn does not exhibit the realism of live competitions that run in real settings. However, it involves a certain degree of realism specified by the instructors in the game's scenario through the definition of the cyberspace including entities that imitate the behavior of real devices. |
| Narrative | ✔ | HackLearn has a narrative defined by the instructor in the '*Description*' attribute of the scenario object. |
| Progression | ✔ | HackLearn supports real-time progression in the game, as single player game. In single player games conflicting and simultaneous actions (Nagarajan et al., 2012) do not occur. |
| Player's identity | ✔ | Learners have a role and a personal profile they maintain. |
| Player's view | ✔ | The view of the game in single player is definite and exclusive for the learner. |
| Interaction | ✗ | HackLearn does not provide interaction with players and non-playable characters |
| **Segment 5: Adaptability** GBL | | |
| Complexity adjustment and tuning of stress levels | ✔ | HackLearn's adaptability facet involves the adjustment of complexity and the tuning of the stress levels in order to optimize the game's effectiveness. To implement game sessions of varying complexities, instructors define a collection of scenarios referring to diverse subjects and associated with various LOs. The scenarios evolve in terms of the number of steps specified, the number of conditions and the number of entities included. To increase or loose the stress levels, the instructors define in the grading schemes the properties related to the time provided to the learners to perform their tasks, the number of actions they have to perform and the support provided by the game. For instance, the presented prototype scenario refers to learners that have a degree in computer science aiming at following a career in cyber security. For this reason, the scenario's complexity is tuned high in order to motivate and challenge the learners. However, the learners are considered inexperienced CtF participants, and thus the scenario has loose time limits and provides strong support to the learners through the provision of hints and teaching materials. |
| Learning history | ✔ | HackLearn stores the learners' learning history in the back-end storage facility (stated in the '*4.4 Runtime*' sub-section) |
| Participant's analysis and available time | ✔ | Instructor takes into account the learner's characteristics (e.g., background, retention, expectations etc.) and the educational context (e.g., available time, budget, presence of an instructor etc.), and forms the appropriate COFELET scenarios for the learner. |
| **Segment 6: Assessment** GBL and CTF | | |
| Feedback GBL | ✔ | HackLearn provides feedback to learners through the textual responses of the terminal and the use of visualizations. Although, it does not include hall of fames and score boards, it displays in the learner's profile the '*LO Degree*' and '*Degree*' metrics, associated with the LOs. |
| Victory conditions GBL | ✔ | HackLearn considers victory conditions in terms of speed (associated with the time passed since the last action), duration (associated with the time passes since the last SEF) and accuracy (associated with the number of actions since the last task). |
| Points GBL | ✔ | HackLearn counts scores and grades |
| Incentives for good practices and disciplinary actions for repeated mistakes GBL | ✔ | HackLearn's instructors define the grading scheme to reward good practices and to penalize unjustified details and repeated errors. |
| Mayer's methodology GBL | ✗ | HackLearn does not employ the Mayer's methodology (Mayer, 2012) |
| Formative and summative assessment CTF | ✔ | HackLearn performs a formative assessment, as it counts and displays the score and informs the learner when a goal is achieved. HackLearn also performs summative assessment, as it records the learning history of learners that available to the instructor. |

| | | |
|---|---|---|
| Assessment features CTF | ✔ | HackLearn's assessment is 'fair', 'objective' and 'comprehensive'. |
| **Segment 7: Issues and Challenges** | | |
| Demands CTF | ✔ | HackLearn demands include the cost of development and the need for cyber security specialists, game developers and instructors. After the creation of the game and the scenarios, the HackLearn learning and training sessions have minimum demands. |
| Frequency of events CTF | ✔ | Learning and training sessions can be repeated very often. |
| Aims CTF | ? | On the contrary with live competitions, HackLearn aims at forming an organized environment which provides possibilities and guidance to learners to adapt by acquiring new KSAs. However, HackLearn is a hacking simulation game that does not take into account operational and maintenance issues such as operational costs of the systems, updates and upgrades, implementation of disaster-recovery policies, backup schemes etc. |
| Diversity of topics CTF | ✔ | Although, the prototype scenario presented in this study is a penetration testing scenario aiming at fostering vulnerability analysis KSAs, the HackLearn can embrace scenarios from different areas of the cyber security domain (e.g., cryptography, cyber threat intelligence etc.). |
| Partial credit CTF | ✔ | HackLearn assessment provides partial credit to the learners even when they do not accomplish a mission but they make some progress towards the scenario's goal (i.e., the capture of flag). |

## 7 Discussion

COFELET framework is a pioneer step towards the analysis and design of effective cyber security game-based approaches. The presented work is a proof of concept for the applicability of the COFELET framework and for the feasibility of the development of effective COFELET games. Sections "*4. Design and Development of COFELET Games*" and "*5. Excerpt of the HackLearn Design*" provide insights into the design of a prototype COFELET game that verify the feasibility of the HackLearn game. Moreover, the above mentioned sections indicate the manner that the foundational elements the COFELET ontology can be adopted in COFELET approaches. Section "*6. Evaluation*" presents the results of the evaluation carried out, providing a preliminary appreciation on the game's effectiveness. The outcomes of the evaluation performed in this work are discussed in the remainder of this section.

The results of our evaluation show that HackLearn embraces many of the features depicted in the GBL concept map. Specifically, it embraces 68/78 characteristics that is 87% of the features of the GBL concept map. Moreover, it embraces several characteristics of the '*Pedagogical Benefits*' and the '*Assessment*' segments of the CtF concept map and it seems has the potential to confront most of the issues and challenges of the field. Consequently, HackLearn promises to enhance the effectiveness of the provided cyber security learning and training. Besides, HackLearn design is based on modern learning theories verifying its pedagogical effectiveness. In particular, HackLearn design is based on the activity theory (through the conformance with the ATMSG model) and it additionally supports a good repertory of learning theories, from behaviorism (e.g., when learners have to improve adopted KSAs in terms of speed and accuracy) to constructivism (e.g., when instructors foster creativity, problem solving and critical thinking capabilities).

Besides, HackLearn assimilates well known cyber security models and standards such as CAPEC and CKC to model learners' actions and strategies towards the unleash of cyber-attacks. The assimilation of these standards is a determining factor in creating in HackLearn a highly organized and parameterized environment where learners' actions are monitored and recorded. Based on the game's observations and recordings the learners' efforts are dynamically assessed through the utilization of efficient assessment schemes defined by the instructor. Subsequently, HackLearn provides the instructors the capability to tune the complexity of the upcoming learning and training sessions by increasing the size of the cyberspace and the number of steps or by making stricter the grade schemes.

HackLearn also promises hands-on cyber security learning and training approaches with lower preparation and running costs compared to live competitions. Once a COFELET game (such as HackLearn) is developed and a collection of scenarios is created, the COFELET compliant cyber security learning and training approaches will have minimum demands. Although the development of COFELET scenarios include a certain degree of logical complexity, the formation of scenarios is facilitated through the reuse of objects stored in the repositories of elements described in the COFELET ontology. The HackLearn has an 'always-on' architecture that allows learner to use it anytime and anywhere. Besides, approaches that adopt game-based learning are more likely to motivate young learners to engage in cyber security increasing the chances to motivate them to chase a career in cyber security.

On the other hand, limitations in the pedagogical effectiveness of HackLearn result from the lack of the multiplayer support as in single-player games learners do not have the chance to work as members of a team, communicate with their teammates, cooperate or compete. In the primary analysis of the presented work the multiplayer support feature was in the plans of the HackLearn game. However, in the first iteration of the study, the inclusion of the multiplayer feature was considered infeasible because it rises very much the complexity of the game's design and the creation of scenarios.

Another issue revealed by the evaluation of HackLearn is that it is a single mode game and it only operates under the umbrella of the hacking simulation game genre. In particular, the game's terminal plays the main role in the UI/UX aspect of the game in which learner has to enter text-based commands. On the contrary, a cyber security game suite including a collection of different genre games, multiple UIs and multiple modes of operation (e.g., certification and competition modes) promises to offer better effectiveness and pedagogical benefits (e.g., enhanced motivation and immersion factors) than HackLearn.

## 8 Conclusions

The COFELET framework is a pioneer step towards the elaboration of a design standard promising to kick start the cyber security education and satisfy the current cyber security needs through the utilization of sound theories and standards, and the adaptation of game-based learning and training approaches.

The contribution of this paper are cause and effect of a design and evaluation of the HackLearn prototype cyber security serious game based on the COFELET framework. As COFELET games are complex systems which embrace models and methodologies from various domains, we provided design recommendations that verify the feasibility of such approaches and facilitate their development. We also proposed an extension of the COFELET ontology describing the foundational elements of a COFELET approach and the manner that these elements can be utilized in the structure of a COFELET game and reused in any COFELET approach. We performed a preliminary evaluation on our work to gain an initial appreciation on the effectiveness of the prototype game. The results gained from our preliminary evaluation are encouraging as HackLearn promises a continuous and pedagogically effective learning and training approach with minimum demands.

Nevertheless, the evaluation revealed some limitations of HackLearn related with the lack of multiplayer support (e.g., team training, collaboration, competition etc.) and the lack of multi-mode operation (e.g., certification mode, competition mode etc.). Though, the multiplayer support is a very important aspect that increases the complexity of the presented approach. For this reason, multiplayer support and multi-mode operation is in the future objectives of the COFELET research, as it will be furtherly studied in the subsequent phases of the current study and it will be included in the descendant games of HackLearn.

In the short-term future plans of the presented study, the HackLearn game implementation will be finalized and the game will be tested for its effectiveness in real-world settings.

## References


Allen P, Straub K. Using Games to Enrich Continuous Cyber Training. Johns Hopkins APL Technical Digest. 2015; 33.2.

Almond R, Steinberg L, Mislevy R. Enhancing the design and delivery of assessment systems: A four-process architecture. Journal of Technology Learning and Assessment. 2002; 1(5).

Anderson L, Krathwohl D, Airasian P, Cruikshank K, Mayer R, Pintrich P, Wittrock M. A taxonomy for learning, teaching and assessing: A revision of Bloom's taxonomy. New York. Longman Publishing; 2001.

Arnab S, Lim T, Carvalho M, Bellotti F, de Freitas S, Louchart S, Suttie N, Berta R, Gloria A. Mapping learning and game mechanics for serious games analysis. British Journal of Educational Technology. 2015;46(2):391-411. https://doi.org/10.1111/bjet.12113.

Ausubel D. The Acquisition and Retention of Knowledge: A Cognitive View. Springer 2000

Bruner J. Acts of Meaning (Vol. 3). Harvard University Press.

Carvalho M, Bellotti F, Berta R, De Gloria A, Sedano C, Hauge J, Hu J, Rauterberg M. An activity theory-based model for serious games analysis and conceptual design. Computers & education. 2015; 87, 166-181. https://doi.org/10.1016/j.compedu.2015.03.023.

Chapman P, Burket J, Brumley D. Pico C: A game-based computer security competition for high school students. In {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 14). 2014.

Chung K, Cohen J. Learning Obstacles in the Capture the Flag Model. In 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14), 2014.

Dewey J. How We Think: A Restatement of the Relation of Reflective Thinking to the Educative Process. USA. Heath & Co Publishers: 1933.

Fellow Traveller. Hacknet. http://www.hacknet-os.com/, 2020 (accessed 30 June 2020).

Hendrix M, Al-Sherbaz A, Bloom V. Game based cyber security training: are serious games suitable for cyber security training? International Journal of Serious Games. 2016;3.1:53-61.

Tommaso D, Di Franco F. (ENISA). Cybersecurity skills development in the EU. https://www.enisa.europa.eu/publications/the-status-of-cyber-security-education-in-the-european-union, 2020 (accessed 30 June 2020).

Glass R. A structure-based critique of contemporary computing research. Journal of Systems and Software. 1995;28(1):3-7.

Greitzer F, Kuchar O, Huston K. Cognitive science implications for enhancing training effectiveness in a serious gaming context. Journal on Educational Resources in Computing. 2007;7.3: 2.

International Information System Security Certification Consortium (ISC). Strategies for Building and Growing Strong Cybersecurity Teams - Cybersecurity Workforce Study. https://www.isc2.org/-/media/ISC2/Research/2019-Cybersecurity-Workforce-Study/ISC2-Cybersecurity-Workforce-Study-2019.ashx, 2019.

Jonassen D, Rohrer-Murphy L. Activity theory as a framework for designing constructivist learning environments. Educational technology research and development. 1999;47.1:61-79.

Katsantonis M, Fouliras P, Mavridis I, Conceptual analysis of cyber security education based on live competitions. 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017, pp. 771-779. https://doi.org/10.1109/EDUCON.2017.7942934.

Katsantonis M, Fouliras P, Mavridis I. Conceptualization of Game Based Approaches for Learning and Training on Cyber Security. Proc. of the 21st Pan-Hellenic Conference on Informatics. 2017. https://doi.org/10.1145/3139367.3139415.

Katsantonis M, Mavridis I. Ontology-Based Modelling for Cyber Security E-Learning and Training. In: Herzog M, Kubincová Z, Han P, Temperini M (Eds.) Advances in Web-Based Learning - ICWL 2019. ICWL 2019. Lecture Notes in Computer Science, Springer, Cham. vol 11841. https://doi.org/10.1007/978-3-030-35758-0_2.

Katsantonis M, Kotini I, Fouliras P, Mavridis I. Conceptual Framework for Developing Cyber Security Serious Games. In 2019 IEEE Global Engineering Education Conference (EDUCON), 2019 IEEE, pp. 872-881., United Arab Emirates, 2019.

Le Compte A, Watson T, Elizondo D. A renewed approach to serious games for cyber security. Cyber Conflict: Architectures in Cyberspace. 2015; 7th International Conference on. IEEE.

Lockheed Martin. Cyber Kill Chain (CKC). https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html, 2014.

Mayer I. Towards a comprehensive methodology for the research and evaluation of serious games. Procedia Computer Science. 2012; 15, 233-247.

MITRE. Common Attack Pattern Enumeration and Classification (CAPEC). https://capec.mitre.org, last accessed, 2020 (accessed 30 June 2020).

Nagarajan A, Allbeck J, Sood A, Janssen T. Exploring game design for cybersecurity training. Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on. IEEE. 2012.

Newhouse W, Keith S, Scribner B, Witte G. National Initiative for Cybersecurity Education (NICE) - Cybersecurity Workforce Framework. NIST Special Publication. 2017; 800, 181.

Oates B. Researching information systems and computing. 1st ed. Sage, 2006.

Risk Based Security. 2020 Q1 Report Data Breach QuickView. https://pages.riskbasedsecurity.com/en/2020-q1-data-breach-quickview-report. 2020 (accessed 30 June 2020).

Sessa V, London M. Continuous learning in organizations: Individual, group, and organizational perspectives. Psychology Press, 2015.

Valve Corporation. Steam (game distribution platform). https://store.steampowered.com/about/, 2020 (accessed 30 June 2020)

Valve Corporation. Hacknet in Steam. https://store.steampowered.com/app/365450/Hacknet/, 2020 (accessed 30 June 2020)

Vygotsky L. Mind in society: The development of higher psychological processes. Harvard University Press; 1978.

Wang R, DeMaria S, Goldberg A, Katz D. A systematic review of serious games in training health care professionals. Simulation in Healthcare. 2016; 11(1), 41-51.