Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

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

Menelaos N. KATSANTONIS[a], *Ioannis MAVRIDIS*[a], *Dimitris GRITZALIS*[b,*]

[a] *Dept. of Applied Informatics, University of Macedonia, Thessaloniki, Greece*
[b] *Dept. of Informatics, Athens University of Economics & Business, Athens, Greece*

ARTICLE INFO

Article history:
Received 13 July 2020
Revised 27 January 2021
Accepted 10 March 2021
Available online 27 March 2021

Keywords:
Cyber security education
Serious games
Design
Evaluation
Ontology
eLearning
Training
COFELET

ABSTRACT

Cyber security game-based learning is a new field that lacks design standards and common methodologies. To this end, the Conceptual Framework for eLearning and Training (COFELET) and the COFELET ontology have been proposed. COFELET is a framework that can be used as a guide for the design and evaluation of effective cyber security learning and training approaches, whereas the COFELET ontology describes the key elements that such approaches should embrace to assimilate well known cyber security threat analysis and modeling standards as the means to create interesting educational experiences. Aiming at providing insights on how COFELET compliant approaches can be developed, we propose the life-cycle of a COFELET game, a blueprint illustrating the design aspects and the course of phases for the development of COFELET compliant games (COFELET games). Besides, an extension of the COFELET ontology is also proposed describing the appropriate elements utilized to develop the learning and the instructional aspects of COFELET games. Based on the life-cycle of a COFELET game and the extended COFELET ontology, we elaborate the design of a prototype hacking simulator COFELET game, called HackLearn. An excerpt of the HackLearn's design is methodically presented along with the analysis elaborated according to the Activity Theory Model for Serious Games (ATMSG) and a set of instances of the COFELET ontology objects. Finally, 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. The results of the evaluation show that HackLearn embraces several features of cyber security game-based learning approaches and they also provide reasons to be optimistic about the effectiveness of the cyber security learning and training that it will deliver.

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). Such challenges are primarily driven by the necessity for more cyber security personnel capable of facing the emerging threats and fighting the cyber criminals in terms of knowledge and competencies. According to the International Information System Security Certification Consortium (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,

* Corresponding author.
E-mail addresses: mkatsantonis@uom.edu.gr (M.N. KATSANTONIS), mavridis@uom.edu.gr (I. MAVRIDIS), dgrit@aueb.gr (D. GRITZALIS).

https://doi.org/10.1016/j.cose.2021.102263

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). However, as game-based cyber security 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 novel multidisciplinary framework embracing a number of appropriate features for the creation of effective game-based cyber security learning and training approaches. Such features are the compliance with serious games development frameworks, the assimilation of well-known cyber security standards, the conformity with sound learning theories and the adoption of the live competitions (e.g., capture the flag (CtF)) characteristics (Katsantonis et al., 2019). Moreover, COFELET employs the COFELET ontology, which provides coherent descriptions of the elements that COFELET compliant games (COFELET games) embrace and their relationships (Katsantonis and Mavridis, 2019). The COFELET ontology describes the elements COFELET compliant approaches must have to model the actions attackers perform to unleash cyber-attacks.

COFELET games are novel multidisciplinary approaches involving considerable degree of complexity for their design and evaluation. Under this prism, the study presented in this manuscript has two research goals. The first is to propose a roadmap to design a COFELET game in the context of its life-cycle and the second is to provide a preliminary evaluation of the COFELET games' potential in delivering effective learning and training outcomes. For the elaboration of the presented work, the design and creation research strategy (Oates, 2005) has been adopted. According to this strategy, the manner that the COFELET ontology key elements can be combined, stored, and reused was initially studied. This process resulted in the definition of new concepts bounded to the existing ones in the COFELET ontology. Subsequently, the design and creation of a prototype COFELET game to apply the COFELET framework was initiated. Then, an iterative research process of analyzing, designing, implementing and testing the structural components of the prototype game (i.e., artefacts) was employed. During this process, the produced artefacts were combined and generalized, and the architectural design of the prototype game, applicable for all COFELET games, was created. To appreciate the effectiveness of the elaborated design, the analysis scheme that we presented in (Katsantonis et al., 2017a) was put into effect.

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; section 4 presents the life-cycle of a COFELET game and a blueprint that facilitates the design and creation of such a game; section 5 states illustrative details on the design of the HackLearn prototype including a description of the prototype HackLearn scenario; section 6 presents the evaluation of the HackLearn design; section 7 discusses the results of the HackLearn design evaluation, and section 8 concludes the paper.

2. Background

2.1. The COFELET Framework

The COFELET framework, presented in (Katsantonis et al., 2019), specifies the main elements that have to be taken into consideration for the design of cyber security serious games, along with the interconnection of these elements in the games' structures. The primary concept of the COFELET framework is the task (represented in Fig. 1 by circles). *Tasks* comprise the actions directed at the fulfilment of a game's goals. The sequences in which tasks could be performed are described in Scenario Execution Flows (*SEFs*). SEFs are proposed to be defined in analogy to attack patterns, e.g., as defined in CAPEC (Mitre, 2020). Learners have to follow one of the predefined SEFs to successfully complete a particular game's scope according to the occurring conditions. *Conditions* represent prerequisites needed to perform tasks. During a game session, the learner's progress is monitored, and their efforts are supported through a teaching contents and hints provisioning system (scaffolding) system. At the end of a game session, the learner's performance is assessed and reviewed, and feedback is provided (i.e., particular achievements) to her. Then, the profile of the learner is updated. In a subsequent game session, a scenario is selected according to the learner's profile and history, as well as the learning objectives, the educational environment and the learning strategy.

To facilitate the analysis of the educational and the gaming components of a cyber security serious game and to link these components to the overall game's learning objectives, 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).

2.2. The COFELET Ontology

The COFELET ontology, presented in (Katsantonis and Mavridis, 2019), provides the analytical descriptions of the key elements COFELET games need to include to interpret cyber-attacks in organized and parameterized learning scenarios. The key elements described in the COFELET ontology are: tasks, conditions, goals and SEFs. *Tasks, conditions* and *goals* are the primary elements of the COFELET ontology, which are represented by quintuple statements (quintuples) of the form `<subject, property, object, source, destination>` (e.g., the task `<File transfer tool - sends - payload file, from learner's host to target host>`). The *subject, object, source* and *destination* in the quintuple statements correspond to the entities that represent distinct concepts that lie in the context of each one game (e.g., agents, hosts, tools, commands). The *property* symbolizes the relation between the games' entities. The primary elements are combined to form the SEFs. *SEFs* are created as representations of attack patterns (e.g., from CAPEC), describing the sequence of tasks that learners follow,

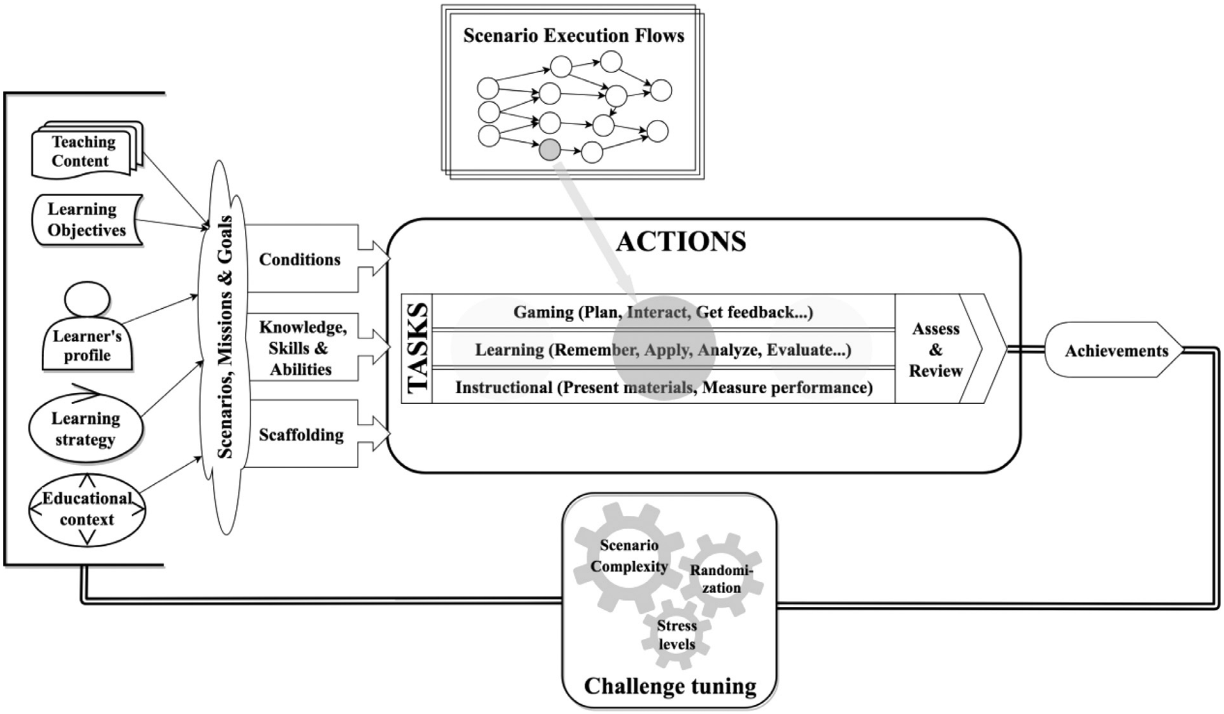

Fig. 1 – The COFELET Framework (Katsantonis et al., 2019).

the conditions that have to occur in the game's context and the goals that have to accomplish.

2.3. Learning Strategies

The COFELET framework has been established on the principles of the activity theory (Jonassen and Rohrer-Murphy, 1999), 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., the COFELET games) as well as the learners' activities in such environments. 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). Therefore, the amount of passive activities such as reading, hearing and watching, is reduced and learners are not yet passive receivers of information as happens with traditional teaching methods (e.g., lectures, workshops, lab sessions) (Dewey, 1933) used in many cyber security education programs (Allen and Straub, 2015). In addition, learners use prior knowledge to experiment with the system and make assumptions and errors that are not traumatic but pedagogically productive (Ausubel, 2000).

The COFELET framework utilizes the *layered learning* approach, presented in (Katsantonis et al., 2019), to feature an effective repertoire of learning strategies. In particular, the COFELET framework assumes modern educational methodologies that comply with modern learning theories and traditional learning and training paradigms. The modern educational methodologies foster critical thinking, problem-solving abilities, and analytical and creative skills by forming realistic environments in which learners solve genuine problems. On the contrary, the COFELET framework also considers traditional learning and training in which learners perform simple activities such as comprehension and recalling of concepts, utilization of tools and practice on tasks. Traditional learning and training paradigms 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 proper tools to dump a computer's memory.

Nevertheless, the COFELET framework adopts the *continuous learning* approach (Sessa and London, 2015) under two perspectives: a) a learner has to try new experiences and challenges; b) a learner has to try known things in new ways. The COFELET framework supports the first perspective by forming scenarios in which the environment becomes increasingly immersive and complex and the learners have to confront new problematic cases tuned to their needs and cognitive levels. On the other hand, under the viewpoint of the second perspective, the COFELET framework 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

The COFELET ontology, presented in (Katsantonis and Mavridis, 2019), aims at analyzing the foundations for the design of COFELET approaches. However, this ontology is restricted in providing analytical descriptions of the key ele-

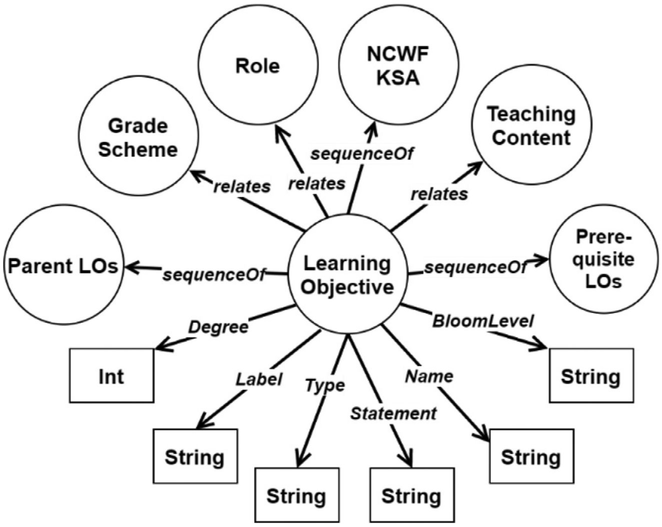

Fig. 2 – COFELET Learning Objective.

ments COFELET games need to comprise in order to simulate cyber-attacks (i.e., primary elements and SEFs). Moreover, the COFELET ontology omits to analyze the elements that facilitate the learning and instructional aspects, a critical façade of the COFELET compliant approaches. For this reason, an extension of the COFELET ontology is proposed in this section to specify the additional elements of *learning objective* (LO), *grade scheme* and *role*. Moreover, the extended COFELET ontology includes revision of the scenario element, presented in (Katsantonis and Mavridis, 2019), to exhibit its associations with the added elements, as well as comprehensive presentation and analysis of all the elements of the extended COFELET ontology.

3.1. *Roles and Learning Objectives*

In general, a role is a position or responsibility that an individual has in an organization or an association. In the COFELET ontology, *roles* are defined based on the cyber security workforce roles found in the National Cybersecurity Workforce Framework (NCWF), which are herein called *parent roles*. The NCWF associates each one role with a set of tasks, knowledge, skills and abilities (KSAs) required by a cyber security professional assigned the role to successfully perform her duties. The COFELET ontology adopts the manner *parent roles* are organized, but it associates them with a sequence of learning objective (LO) elements, which herein are called *parent LOs* and they are defined by utilizing the NCWF's KSAs. In the COFELET ontology, the LO elements are represented by the *LO* class (Fig. 2).

A Role element is represented by the Role class. The *Role* class contains a role description and a sequence of LOs. The *Role* class associates the LOs it embraces with the following data properties:

1. *LO degree*: demonstrates the degree the learner possesses the parent LOs associated to her assigned role. The *LO degree* property can be used as a dynamic metric that changes overtime. Its value increases by the amount specified in the *Degree* attribute of the associated LO class (defined in Table 1) when the LO possession is achieved, and decreases by the value of the decay factor property after a period of time specified by the value of the *inactivity* property.
2. *inactivity*: indicates the time period of learner's inactivity.
3. *decay factor*: specifies the amount of decreasing the value of the *LO degree* property.
4. *last update*: is the date and time of the last change of the value of the *LO degree* property.

The *LO* class contains several object and data properties listed in Table 1.

As cyber security is a rapidly changing field, new LO objects is necessary to be created on demand (Katsantonis et al., 2019). In such cases, instructors can define LOs that might not be based on *parent LOs*. Likewise, new roles can be created and associated with the new LOs. In a COFELET game, a new role can be created as a combination of two (or more) COFELET roles, and it can be associated with a sequence of LOs consisting of existing and new LOs. For example, the creation of the role of *Penetration tester* is the result of combining the *Vulnerability Assessment Analyst* and the *Target Network Analyst* roles (parent roles). The *Penetration tester* inherits the LOs from its parent roles (resulting so in a hierarchy of roles), which are related to the knowledge of the penetration testing principles and techniques, the knowledge of threats and cyber-attacks and the proficient use of penetration testing tools.

3.2. *Scenarios*

A Scenario element contains the appropriate information for the setup of a game session and it consists of three parts (Fig. 3):

1. *Attributes*: the name, the label, the description, the narration and the difficulty level of the scenario.
2. *Cyberspace*: a collection of conditions (i.e., scenario's preconditions) that are in effect when the game session starts, and a set of entities forming the scenario's cyberspace.

Table 1 – The attributes of LO class.

Attribute	Rational
Name	the LO's unique name
Label	the LO's user-friendly name
Statement	the LO's statement in the form <Learner - Property - Object>
Type	indicates whether the LO is task, knowledge, skill or ability
Degree	is associated with the *LO degree* property defined in the Role class. It indicates the amount of increasing the value of the *LO degree* property after a LO possession achievement. Its value is specified by an instructor based on the scenario's complexity and the mission's difficulty
Role	the associated role(s)
BloomLevel	the mapping level in the Bloom's taxonomy
Prerequisite LOs	a sequence of prerequisite LOs the learner has to possess
Parent LOs	a list of parent LOs utilized for the definition of the LO
Teaching Content	texts, figures and videos presenting the KSAs aimed to be transferred to the learners
Grade Scheme	a rubric according to which the learner's efforts and progress are assessed

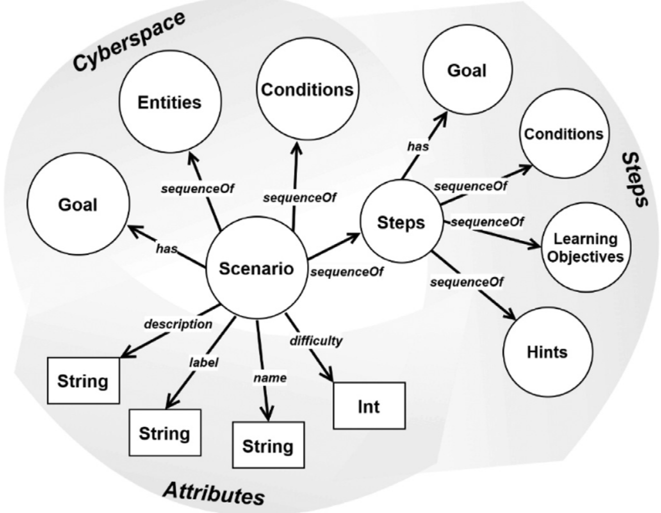

Fig. 3 – COFELET Scenario.

3. *Steps*: a sequence of steps corresponding to the stages of a mission. Each step contains a sub-goal, a set of conditions (e.g., pre-conditions and post-conditions), a set of LOs and a sequence of hints. A Hint element consists of the *text* attribute that is the suggestion provided to help learners achieve the game goals and the *time* attribute that denotes the period after which learners are notified to read the hint.

3.3. *Grade Scheme*

The Grade Scheme element is associated with a LO element and a LO element is associated with a Scenario's step (presented in sub-section '3.2 Scenarios'). Thus, a grade scheme is applied to assess the learner's efforts at the end of a step. The Grade Scheme class consists of the following attributes:

1. *grade*: specifies the points assigned to the learner. The grade's value calculation is based on the values of the attributes 2 to 5 presented below.
2. *assessed*: denotes how many times the LO associated with the current grade scheme has been assessed. The value of the attribute is retrieved in the learner's learning history.
3. *hints*: logs the number of hints provided to the learner with respect to the number of available hints in the associated step.
4. *time*: records the time it took the learner to complete the associated step and achieve the possession of the LO.
5. *actions*: logs the number of actions the learner performed in the step to achieve the possession of the LO.
6. *score*: specifies the signed percentage factor applied to the value of the *Degree* attribute specified in the LOs. The result determines the amount of value affecting the *LO Degree* attribute of the Role class.

In many cases, the *score* and *grade* attributes have the same 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 (Nagarajan et al., 2012) when learners do not achieve the game's LOs, even when the LO possession has been exercised a number of times. On the contrary, the value range of the *grade* attribute is from 1 to 100.

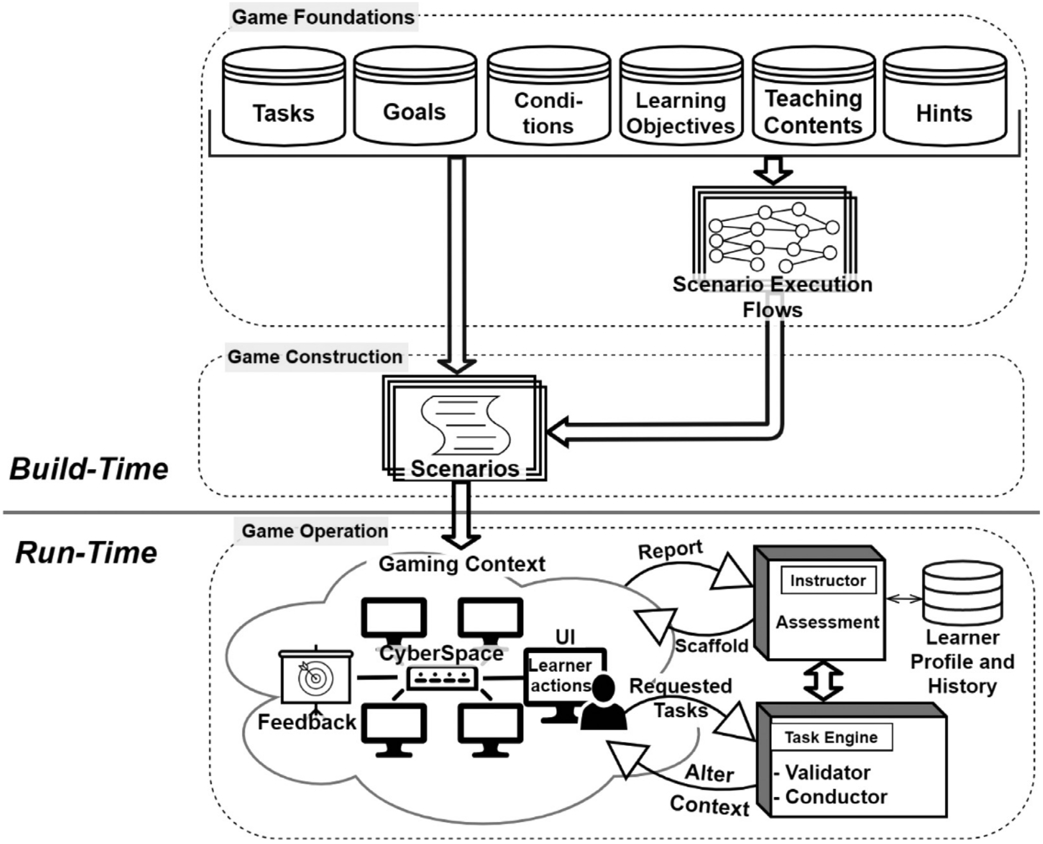

Fig. 4 – The COFELET game life-cycle.

4. The COFELET Game Life-Cycle

In this section, the COFELET game life-cycle (illustrated in Fig. 4) is presented as a mean to exhibit how the game's major components and the elements of the COFELET ontology are organized in the structure of a COFELET game. Moreover, the main actors involved in the life-cycle of a COFELET game and their cooperation is presented.

The life-cycle of a COFELET game consists of two phases: *Run-Time* and *Build-Time*. The *Build-Time* phase contains two sub-phases: *Game Foundations* and *Game Construction*. In the *Game Foundations* sub-phase, the key elements described in the COFELET ontology are created, whereas in the *Game Construction* sub-phase the COFELET scenario elements are formed. In the *Run-Time* phase, the major components of the game are depicted along with the functions they perform and their interconnections (*Game Operation*).

4.1. *Actors*

The use case diagram in Fig. 5 depicts of the actors' involvement in the COFELET game life-cycle. Specifically, the actors involved are: game developers, cyber security specialists, instructors and learners. *Game developers* work at *Game Foundations* and *Game Construction* sub-phases to create the games by implementing the designs of the cyber security specialists and the instructors. *Cyber security specialists* have deep knowledge of cyber security methodologies and models (e.g., the CKC model (Lockheed Martin, 2014)) and utilize the COFELET ontology at the *Game Construction* sub-phase to design the key elements that will be interpreted in the particular games (e.g., SEFs). *Instructors* are educators, aware of the parent roles and the corresponding KSAs, who complement the work of cyber security specialists at Game Construction sub-phase by adding the elements that determine the learning and instructional perspectives of COFELET games (i.e., LOs, hints and teaching content). Instructors also can cooperate with game developers at Game Construction phase to create or edit game scenarios. Conclusively, *learners* are the final recipients of the COFELET games using them at the Game Operation phase.

4.2. *Build-Time*

During the *Game Foundations* sub-phase, the repositories of key elements, depicted in the upper part of Fig. 4, are created and they are stored in a manner that facilitates their adoption in diverse games and educational contexts.

During the *Game Construction* sub-phase, instructors create the COFELET scenarios by utilizing the key elements stored in the repositories. Scenarios describe in-game entities by providing the necessary properties for imitating the behavior of real devices (e.g., networked hosts), including some attributes with randomized values that change from session to session. Scenarios can also contain additional elements when instructors need to add extra functionalities and features. In such

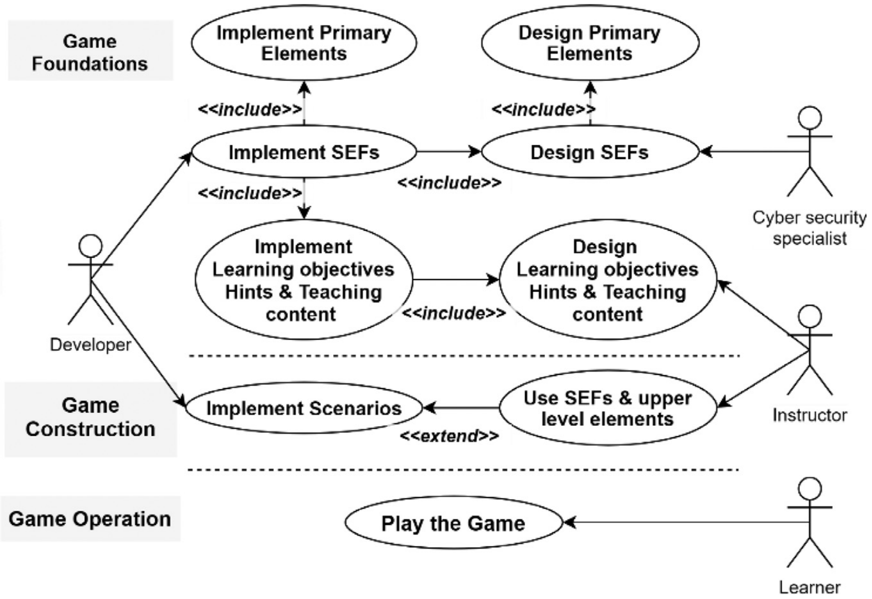

Fig. 5 – Use case of the actors involved in the life-cycle of a COFELET game.

cases, instructors need to cooperate with game developers during the Game Construction sub-phase. For example, a scenario can include the *Question* elements, which are additional elements representing the questions issued during the game play. The *Question* elements must be explicitly associated with particular LO elements and they regard cyber security concepts (e.g., employed attack patterns).

4.3. *Run-Time*

The Run-Time phase of the COFELET game life-cycle depicts the following components of a COFELET game:

- *Gaming Context*: contains the user interface façade (UI) and the game's Cyberspace. The game's Cyberspace is the virtual environment in which learners perform their actions and they unleash their cyber-attacks. It 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 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 game's 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.
- *Task Engine*: is a task operator that conducts the performed tasks and provides feedback to the learner through the game's Cyberspace. It consists of a *Validator* and a *Conductor*. The *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. The *Conductor* virtually executes a task and checks whether its execution provokes the fulfillment of a goal or a mission. The *Conductor* also sets the post conditions of the executed task and communicates with the *Gaming Context* and *Instructor* components.
- *Instructor*: assesses the learning session and scaffolds the learner's efforts. Specifically, the Instructor component:
  - monitors the learner's progress and acquires the necessary information from the *Task Engine* and the *Gaming Context*. The details acquired include the learner's actions, the tasks performed, the goals achieved and the currently applied SEF by the learner.
  - manages the appropriate key elements such as the scenario's hints, the teaching contents and the LOs whose possession the learner has to achieve.
  - 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 history of learning and training.
  - scaffolds the learner's efforts through the provision of hints and teaching contents that are associated with the LOs whose possessions the learner has to achieve. For example, it counts the game play time period and it monitors the learner's progress. Whenever, the game play time period is beyond a time threshold specified by the instructor in the game's scenario (i.e., in the *time* attribute of a hint element specified in subsection '3.2 Scenarios'), the learner is notified and the appropriate hint(s) are shown to her.
  - assesses the learner's fulfillment by applying a grading scheme specified by the instructor actor in the scenario (i.e., the *Grade Scheme* objects). Subsequently, the assessment details are stored in the back-end storage facility and the learner's profile is updated.

The *Run-Time* phase cycle exhibits the manner according to which COFELET games realize the perspectives of:

- Gaming: Particularly, a COFELET game renders the learner actions in two sites: in the game's Cyberspace and in the Task Engine. The game's Cyberspace emulates the real world and it interprets the learner actions under the gaming perspective. For example, a COFELET game can imitate 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.
- Learning: 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, which are explicitly related with the learning and the instructional aspects of the game (e.g., the LOs, the hints and the teaching materials). In such way, a game can translate the learner's actions to accomplishments of LOs possession related with obtaining the required KSAs.
- Instructional: 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 assessing the learners' efforts, provisioning hints and teaching contents explicitly related with the learners' tasks.

5. HackLearn

In this section, a design excerpt of a COFELET game, called HackLearn, is presented. Initially, the characteristics of HackLearn are presented including its genre, the features it adopts from the CtF competitions and the characteristics that distinguish it from other cyber security game-based learning approaches. Subsequently, the application of the ATMSG model is demonstrated to exhibit the HackLearn's game flow and to analyze it under the gaming, the learning and the instructional perspectives. Finally, the HackLearn prototype scenario is presented that puts together the elements discussed in previous sections 3 and 4. Specifically, the HackLearn prototype scenario:

- allows learners apply an attack based on the CKC model (CKC attack);
- exemplifies the COFELET game life-cycle by providing details on the manner its components interact with each other,
- exhibits how the layered learning and the continuous learning approaches (as analyzed in sub-section 2.3) are employed;
- demonstrates how the new elements of the extended COFELET ontology (e.g., Roles, LOs, and Grade Schemes) facilitate the learning and the instructional aspects of the game.

5.1. *HackLearn's Characteristics*

Cyber security educational games are already used for teaching various cyber security topics in miscellaneous contexts. Researchers in (Hendrix et al., 2016) reviewed and categorized cyber security serious games according to their game type, the methodology they apply, the cyber security topics they aim to teach, the target audience and the evaluation performed. However, in their literature review, very few cyber security serious games found with target audience cyber security professionals and university students. Besides, none of the identified games offer opportunities for hands on experiences and practices, such as the use of cyber security tools to unleash cyber-attacks. On the other hand, commercial hacking simulation games 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 1.000.000 to 2.000.000 owners, more than 70.000 followers and more than 10.000 positive comments (Valve, 2020b). However, the hitherto known hacking simulation games are not included in cyber security education research ((Hendrix et al., 2016), (Katsantonis et al., 2017a), (Mostafa & Faragallah, 2019)), as they are out of the scope of such a research. The reason for this is that they are commercial games that do not clearly have learning and training as their primary objective, rather fun and entertainment. Besides, they are not designed to operate as learning or training tools in educational contexts, they have different target audience than cyber security educational games, they do not have clear learning objectives and often they are not based on authentic cyber security topics (e.g., they use logical puzzles).

HackLearn is a cyber security serious game that can be mainly included in the hacking simulation game genre (i.e., hacking simulator), as it is a cyber security educational tool that adopts the characteristics of the hacking simulators genre. Specifically, HackLearn includes a Unix-like terminal in which players utilize emulations of real-world tools by typing and executing text-based commands (e.g., nmap, base64, whoami); simulations of cyber-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; and cyber security missions based on scenarios.

HackLearn draws many elements from CtF competitions, designed for educational purposes, as the learners unleash cyber-attacks during the game-play, they collect flags and points, they exercise their knowledge and skills, and they try to beat the clock. Though, HackLearn is not a CtF competition exercise or game, it adopts CtF competitions' features and it also tries to overcome the CtF's limitations reviewed and analyzed in (Katsantonis et al., 2017a). Moreover, HackLearn diverges from CtF games because such approaches provide cyber-security hands on experiences in an unstructured and self-directed manner (Katsantonis et al., 2017a). On the contrary, HackLearn forms a highly organized and parameterized environment that dynamically monitors learners' actions, evaluates their progress and it scaffolds their efforts. To do so, it utilizes the COFELET framework and the COFELET ontology to model strategies and cyber-attacks and to support the learning and instructional aspects of the game.

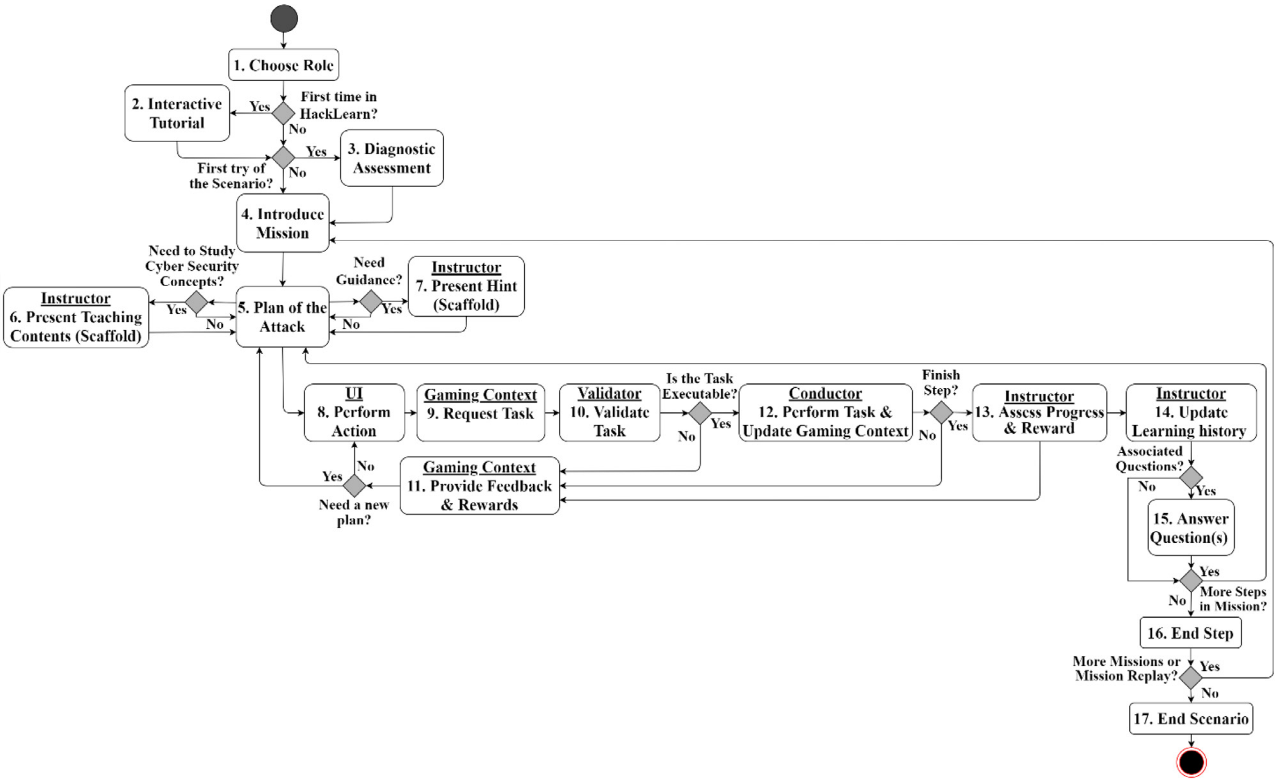

Fig. 6 – HackLearn's Sequence Diagram.

5.2. *HackLearn's Analysis*

In this section, the ATMSG model is used to describe the HackLearn's game flow and demonstrate the systematic organization of the HackLearn's serious game components. According to the ATMSG model approach, after the identification of the HackLearn's activities, the game components are initially presented in a UML activity diagram, the game's sequence diagram. The diagram in Fig. 6 shows HackLearn's sequence diagram presenting the game's components and the manner that they are interconnected throughout the game. The HackLearn's sequence diagram consists of seventeen (17) components, some of which embrace the functionalities of the *Instructor, Validator, Conductor* and *Gaming Context* components of the COFELET game life-cycle (described in sub-section 4.3).

Table 2 exhibits the analysis elaborated upon the components of the HackLearn's game sequence diagram. Particularly, Table 2 identifies HackLearn's components and classifies them in the perspectives of gaming, learning and instructional according to the activities they embrace. In Table 2 is also specified for each component the *actions* performed in the game, the *tools* that make these actions possible and the *goals* as the objectives that will be achieved after the accomplishment of the *actions*.

For the design of the HackLearn's components, the elements of the ATMSG taxonomy for serious game components (Carvalho et al., 2015) were utilized. Subsequently, detailed descriptions of the HackLearn's components were derived (presented in Table 3), including further details on the activities taking place in HackLearn, the game's *tools* utilized (e.g., the terminal, the progress bar) and the purpose driving these activities. For brevity, the components already presented in subsection 4.3 are not included in Tables 2 and 3, though they are illustrated and specified in the HackLearn's sequence diagram (depicted in Fig. 6).

5.3. *Prototype Scenario*

The prototype scenario presented in this section is inspired from the penetration testing field of cyber security as a multistep complex case of scenario. It aims at training computer professionals with a strong background in cyber security aiming at acquiring knowledge and competencies of the 'Vulnerability Assessment Analyst' role of the NCWF.

The narration of the prototype scenario is recorded in the description attribute of the scenario object and it 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 LOs defined in the prototype scenario are listed in Table 4; for the sake of brevity only the LO statement value is shown along with the matching Bloom level and a reference Code. Table 5 provides a list of the scenario's steps learners follow to apply a CKC attack along with a description of each one step, its corresponding LOs and a reference Code. Fig. 7 provides a diagram of the scenario's cyberspace depicting the scenario's entities with which learners interact to achieve the game's goals.

Most of the LOs presented in Table 4 belong to the application level of the Bloom taxonomy, as the prototype scenario mainly exercises skills in penetration testing. However, the

Table 2 – HackLearn’s Serious Game Components.

		1. Choose Role	2. Interactive tutorial	3. Diagnostic Assessment	4. Introduce Mission	5. Plan of the Attack	Instructor	8. Perform Action	11. Gaming Context		15. Answer Question(s)
							6-7. Scaffold		Feedback	Reward	
Gaming	Actions	Customize	Obtain help	-	Read Story	Plan/Strategy, Match	Obtain help	Create, Generate	Read Information	See Performance Evaluation	-
	Tools	Role	Tutorial	-	Story	Information	Advice and Assistance, Information	2D space, Time pressure	Complete Information	Progress bar, Points, Role/Virtual skills, Status level, Information	-
	Goals	Configure game	Learn to use interface	-	Get acquainted with story (and mission)	Complete quest & side quests	Complete side quests, form/discover goal	Complete quest & side quests	Collect Information	Maximize Performance	-
Learning	Actions	-	Observe, Practice	-	Observe, Identify	Observe, Identify Hypothesize, Combine, Plan, Restate	Read, Find more information about	Apply, Recall, Repeat	Verify, Find More Information About	Verify, Review	Describe, Explain, Summarize
	Tools	-	Tips, Tasks	-	Problem, Challenge	Creations, Inventions	Texts, Information, Tips, Definitions	Simulator, Experiment	Texts, Information, Illustrations (text images)	Information, Graphics	Test, Definitions, Conclusions
	Goals	-	Apply, Remember	-	Understand, Analyze	Active Experimentation, Abstract Conceptualization	Remember, Understand	Apply, Concrete Experience	Remember, Understand, Reflective Observation	Understand, Reflective Observation	Understand, Reflective Observation
Intrinsic Instruction	Actions	-	Demonstrate, Scaffold	Present Quiz	Tell Story, Present Problem	Repetition, Scaffold	Scaffold, Present material	Reward good performance, Repetition	Demonstrate	Qualitatively assess performance	Present Quiz
	Tools	-	Tips	Question and Answers	Story	Information, Multiple choices, Limited set of choices	Tips/Assistance, Help text	Performance measures, Multiple chances,	Tips, Warning messages	Performance measures	Questions & Answers
	Goals	-	Provide learning guidance	Assess Performance (Initial knowledge)	Inform Learner, Gain Attention	Provide learning guidance	Stimulate recall of prior knowledge, Provide learning guidance	Elicit performance	Provide feedback	Assess performance, Provide feedback	Stimulate recall of prior learning

Table 3 – Detailed Description of HackLearn's Serious Game Components.

Node	Gaming	Learning	Intrinsic Instruction
1. Choose Role	The learner chooses a game role associated with LOs whose possession must be achieved. LOs are also associated with the scenarios to be assigned to the learner.		
2. Interactive tutorial	At the beginning of a game session, the learner encounters pop-up messages that explain the basic features of the UI and require her to perform certain actions (e.g., enter a help command in HackLearn's terminal, use the toolbar).	The learner reads the text of the pop-up messages, applies basic UI actions and observes the results of her actions. In such way, she becomes remember basic UI actions when the mission starts.	The tips in the pop-up messages and the related graphics provide learning guidance by directing the learner to perform basic UI actions and acquire the necessary skills in using HackLearn.
3. Diagnostic Assessment			Participants complete a questioner to determine their initial knowledge prior to the game session.
4. Introduce Mission	It demonstrates the mission (i.e., the narration attribute of the scenario object) stating the story and the goals of the mission (e.g., to capture the flag).	The learner reads scenario's narration to understand the problem, to analyze it and to identify objectives and clues.	It informs about the game's objectives and it uses storytelling to grab learner's attention and to motivate her.
5. Plan of the Attack	The learner envisages an attack, decides on the strategy she will employ to reach the game's goals and selects a SEF to apply.	The learner observes the mission's objectives and the scenario's cyberspace, combines information, makes hypotheses and elaborates the plan of the attack (active experimentation). If the plan fails, the learner comes to conclusions (abstract conceptualization) and repeats the process by utilizing what she has learned from her previous experiences.	It scaffolds the efforts of novice learners by displaying the available SEFs and tools or by requiring learner to select only applicable SEFs.
6-7. Scaffold	The learner obtains short suggestions for the completion of the current step (e.g., applicable SEFs, usage of tools, command syntax,) and teaching materials explaining the cyber security concepts involved (e.g., attack patterns, techniques, tools etc.).	The learner studies and recalls the SEFs, the tools and their applicability in the cyberspace of the HackLearn's scenario.	It scaffolds learner's efforts (e.g., to choose the proper SEF or use the appropriate tools) and it keeps learner motivated. It gradually supports player's efforts by successively providing hints that merely reveal part of the solution and increase game's support.
8. Perform Action	The learner performs actions by entering Unix-like textual commands in the HackLearn's terminal.	The learner repeats commands to develop patterns of tool usage under the appropriate conditions.	It measures the number of tries and it counts the time taken to successfully perform a *Task* (i.e., action directed at the fulfilment of *Goals*).
11. Gaming Context: Feedback & Reward	It provides: 1) feedback mainly through the game's terminal, 2) rewards through the UI elements (e.g., score) and the game's graphics (i.e., progress bar) informing the learner about a successful or unsuccessful execution of a task and an attack.	The learner realizes the results of her efforts, reflects on the actions performed and envisages how to improve performance.	It provides points as incentive for good practices or penalty points (or no points) as disciplinary actions for repeated mistakes. Warning messages and tips inform the learner about the *Conditions* ignored or overlooked.
15. Answer Question(s)		The learner answers questions that make her to reflect on her achievements and describe or discuss concepts, techniques, methodologies etc.	It assesses the learner's knowledge gain

Table 4 – The LOs defined in the prototype 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 (i.e., a service that provides a collection of vulnerabilities and code exploits)
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

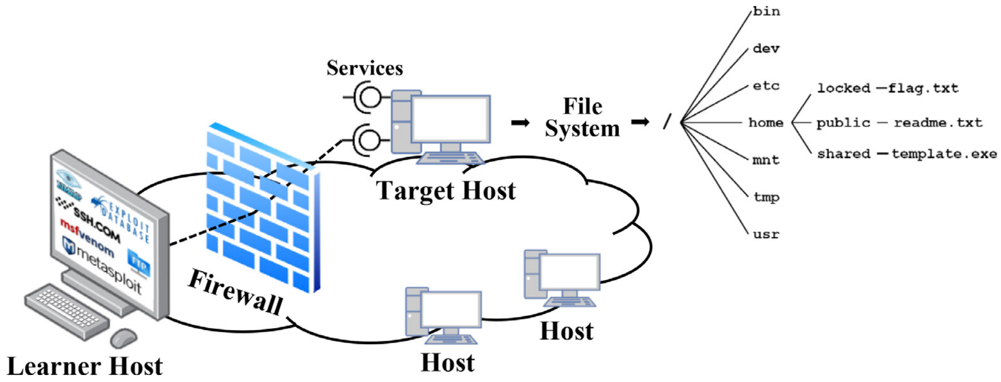

Fig. 7 – Cyberspace of the prototype scenario.

LOs L3 of step S3, L6 of step S4 and L12 of step S8 belong to the higher COFELET layers (Katsantonis et al., 2019) because they require deep knowledge and competencies. Specifically, in step S3, learner has to appraise which host and service is most likely to be vulnerable, otherwise she will end up searching all the services in order to possible vulnerabilities. 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, L13 belongs to the low COFELET layers, as the learner distinguishes the port states. L13 is associated with the question '*Is a filtered target port considered opened or closed?*' prompted in S2. The prototype scenario contains several low level LOs associated with questions prompted during the game-play (they are not listed in Table 4 for brevity).

Table 6 provides a full description of L2's rational and attributes, associated with S2.

The prototype scenario's cyberspace (Fig. 7) is a collection of entity objects with the appropriate attributes and functionality to imitate the behavior of the real devices. Fig. 8 depicts a UML class diagram of the prototype scenario's entities.

5.4. *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

Table 5 – The steps of the prototype scenario.

Code	Label	Description	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

diagram in Fig. 9 shows the manner that the components (depicted in the Run-Time phase of Fig. 4 in section 4), interact to perform command execution actions. Once the learner enters a command, the terminal renders the command's arguments and passes the command to the appropriate built-in tool. 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, performs the command and alters the gaming context. Finally, the learner receives feedback from the UI in various forms (e.g., scores, visualizations, sounds etc.) and from the terminal in textual form.

6. Evaluation

6.1. *Scheme*

The evaluation of the presented HackLearn's design is based on the analysis and evaluation scheme proposed in (Katsantonis et al., 2017a) for conducting preliminary evaluations on new live competition approaches.

Specifically, the evaluation scheme employs a concept map of game-based learning approaches key elements (GBL concept map) depicted in Fig. 10 and a categorization of challenges as an assessment tool for the deduction of assumptions regarding the feasibility and the educational impact of new game-based learning and training approaches as well as the effectiveness of these approaches in coping with the identified challenges.

As HackLearn draws many elements from the live competitions domain, the evaluation scheme also utilizes the concept map of live competitions' technological and pedagogical characteristics (CtFs concept map) depicted in Fig. 11. Additionally, it employs the identified problems and issues of the field presented in (Katsantonis et al., 2017a) and in (Katsantonis et al., 2019).

Particularly, the GBL concept map has the main role in the evaluation process as it encompasses the characteristics of cyber security game-based learning approaches found in literature; the CtFs concept map has a secondary role because only the *Pedagogical Benefits* and the *Assessment* segments are utilized as consistent with the COFELET framework.

6.2. *Results*

In this sub-section, the design of HackLearn is put on the test of the evaluation scheme stated above. In particular, HackLearn is resolved into its elements, and its pedagogical effectiveness is appreciated by comparing HackLearn's characteristics with characteristics of concept maps and in accordance to the issues and challenges it tries to confront.

The results of HackLearn's evaluation are presented in Table 7. Table 7 consists of seven parts; six parts in analogy

Table 6 – L2 description.

Attribute	Value and Rational
Statement	Learner utilizes network analysis tools (i.e., port scanners) to determine the status of ports on targets'
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
NCWF 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: GradeScheme1. "*times assessed='0-1', hints='0', time='1-50', actions='0-3', score='100', grade='100'* ". GradeScheme2. "*times assessed='4-6', hints='1-4', time='1-50', actions='0-3', score='-10', grade='1'* " 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). 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 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 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 Hint 4: 'In the terminal issue the command: *nmap –sS |target|*', time='*120*'seconds

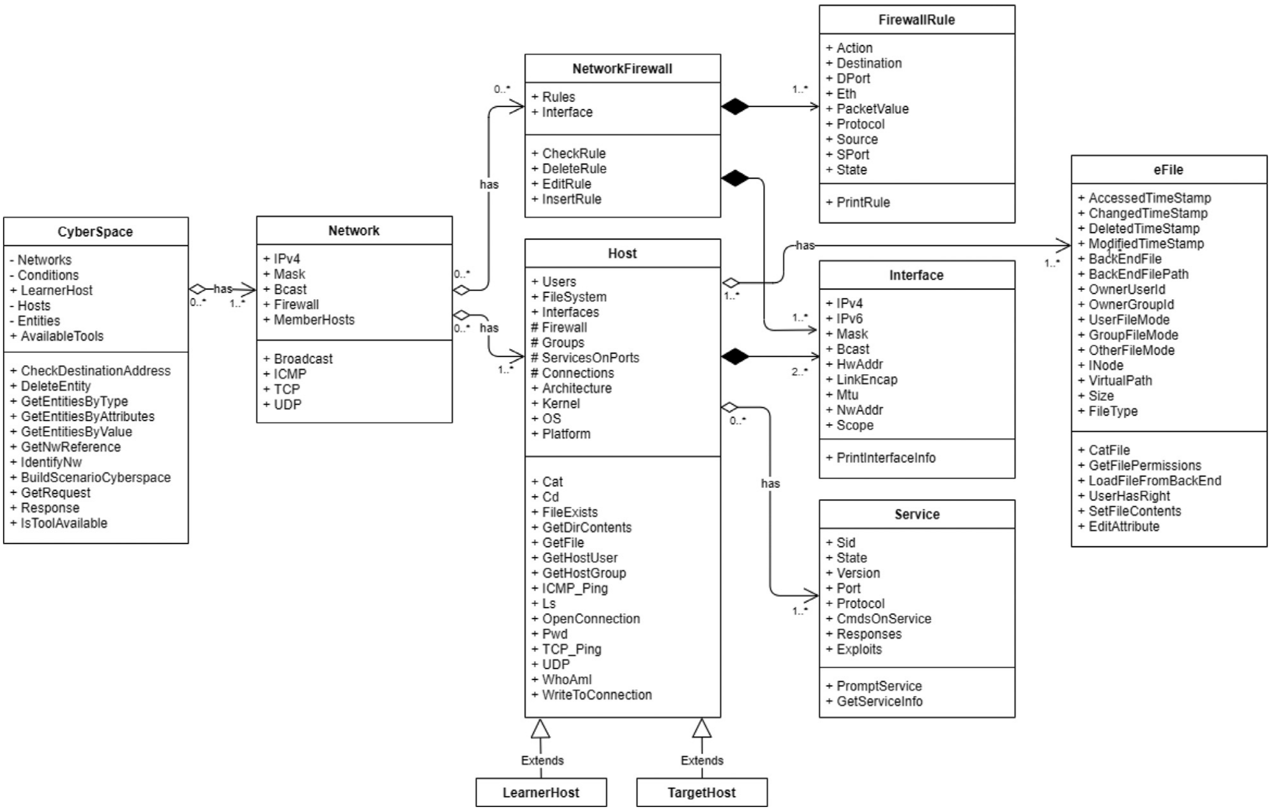

Fig. 8 – Class diagram of the prototype scenario's entities.

Table 7 – HackLearn’s evaluation.

Characteristics	Support	Rational
Segment 1: Pedagogical Considerations $_{\text{GBL}}$ and Pedagogical Benefits $_{\text{CTF}}$		
Cognitive learning $_{\text{GBL}}$	√	HackLearn is based on the cognitive learning theories, as it constitutes an educational environment where learners can 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 a CKC attack.
Creativity $_{\text{GBL}}$	√	In HackLearn, instructors define scenarios in which learners 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 $_{\text{GBL}}$	?	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 $_{\text{GBL}}$	√	HackLearn implements a continuous learning approach, as the game is ‘always-on’ providing the means for organizing learning sessions repeatedly. In learning sessions, learners acquire new KSAs or 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 whose possession has not been achieved for a specified period (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 $_{\text{CTF}}$	×	As opposed to live competitions which promote self-directed learning, HackLearn promotes instructional learning.
Exercise of knowledge, skills and abilities $_{\text{CTF}}$	√	In HackLearn, learners exercise techniques and basic skills such as discovering live hosts in a network (in steps S1), scanning the target’s ports (in step S2) and creating a weapon payload file (in step S6).
Collaboration $_{\text{CTF}}$	×	HackLearn is a single player game and lacks the promotion of collaboration among learners in the context of the game.
Segment 2: Learning Outcomes $_{\text{GBL}}$		
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 4.3).
Learning outcomes show purpose and they are measurable	√	HackLearn’s LOs are based on the parent 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, HackLearn 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 $_{\text{GBL}}$		
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 scenarios.
Manage portfolios of learning objects	√	HackLearn’s repositories can be considered as 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. Thus, it embraces one of the three modes of operations.

(continued on next page)

Table 7 – (*continued*)

Characteristics	Support	Rational
Incorporation of various games	r	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 sub-section 5.1).
Team training	r	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 sub-section 4.3)
Participant's analysis and available time	√	The instructor considers 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 scenarios for the learner.
Segment 6: Assessment GBL and CTF		
Feedback GBL	√	HackLearn provides feedback to learners through the textual responses of the terminal, the use of visualizations and the providence of a score leader board. Besides, HackLearn displays in the learner's profile the *LO Degree* and *Degree* metrics, associated with the achieved LOs possession.
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.

(*continued on next page*)

Table 7 – (*continued*)

Characteristics	Support	Rational
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 $_{CTF}$	?	As opposed to 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).

to equal number of 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) and one part for the issues and challenges of the field. The column *Characteristics* contains the characteristics of both the GBL and CtFs concept maps. For brevity, the sibling characteristics (e.g., formative and summative assessment) are presented in the same rows of Table 7. The segments and the characteristics listed in Table 7 have subscripts indicating the concept map they are adopted from (i.e., *GBL* and *CTF*, accordingly). The column *Support* specifies whether the characteristic is supported (symbol '√'), not supported (symbol '×') or merely supported ('?'), whereas the column *Rational* explains the rationale of the *Support* specification.

Table 7 shows that HackLearn embraces 68 out of 78 characteristics (i.e., that is 87%) of the GBL and CtF concept maps, from which 5 characteristics (i.e., *engagement, immersion, motivation* and *fun* of segment 1 and *realism* of segment 4) are merely supported. On the other hand, from the 10 characteristics not supported by HackLearn, 5 characteristics are associated with the lack of multiplayer support (i.e., *collaboration* of segment 1, *multiplayer* and *competition mode* of segment 3, *team training* and *player interaction* of segment 4). HackLearn embraces 14 of the 17 characteristics of the *Pedagogical Benefits* and *Assessment* segments (i.e., 82%), though it does not support the *self-directed learning* characteristic and the *collaboration* and *teams* characteristics of the *Pedagogical Benefits* segment that are also included in the GBL concept map. On the

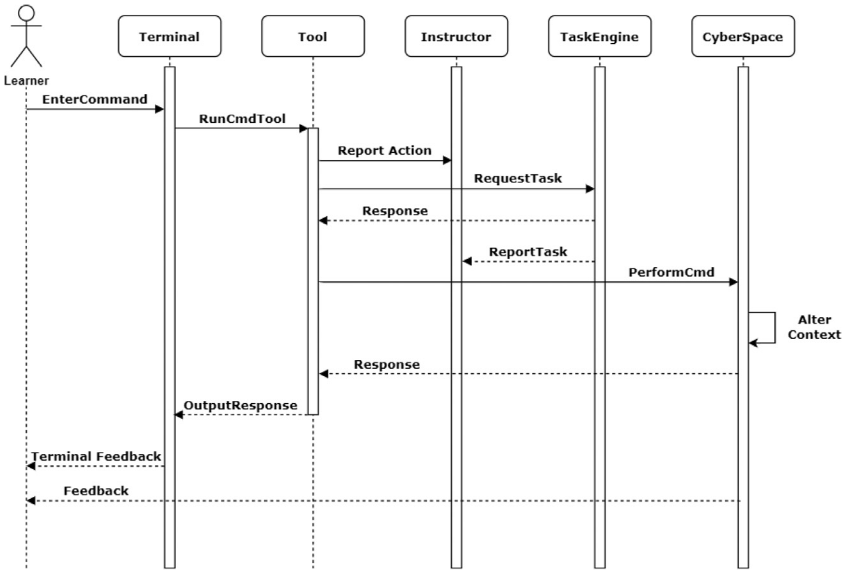

Fig. 9 – Command execution sequence diagram.

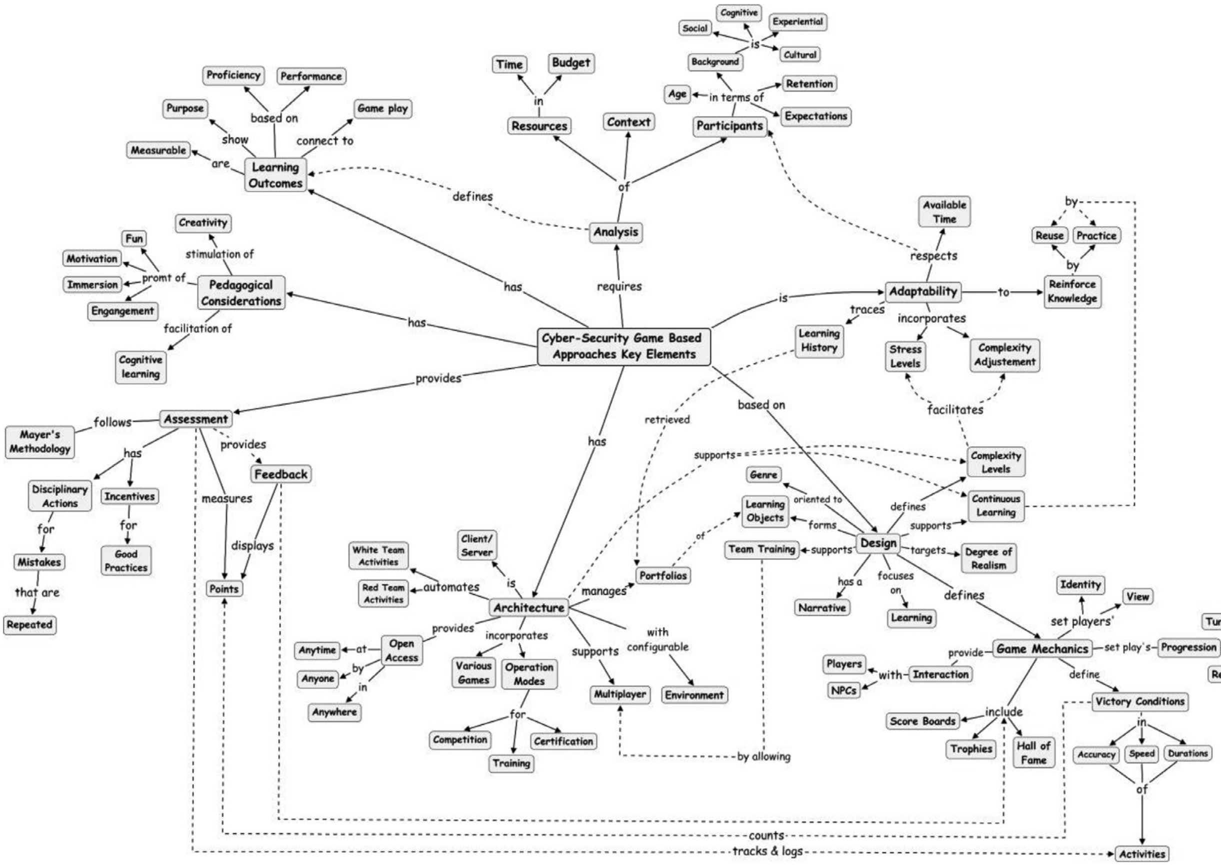

Fig. 10 – GBL concept map (Katsantonis et al., 2017b).

contrary, HackLearn supports all the characteristics of the *Assessment* segment, yet it does not require the learner to write up reports or the presence of a supervisor to perform the summative assessment. Finally, the *Issues and Challenges* segment shows that HackLearn can confront all the challenges and issues identified in live competitions field, apart from the challenge that it does not include and realistically present the views of systems associated with the operational costs, the update and back up policies of systems, etc.

7. Discussion

The results of the evaluation presented in section 6 allow a good appreciation of HackLearn's learning and training effectiveness, as most of the key elements proposed in (Katsantonis et al.,2017b) and in (Katsantonis et al., 2017a) are embraced. Specifically, HackLearn embraces several pedagogical characteristics (listed and analyzed in segments 1 and 5 of Table 7) including the conformance with modern learning theories (presented in sub-section 2.2) that verify its effectiveness. HackLearn's design is based on the activity theory (through the conformance with the ATMSG model) and it additionally supports a 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).

Moreover, the evaluation of HackLearn's design shows a valuable perspective in the assessment of learners' efforts. That is because HackLearn assimilates well known cyber security models and standards such as CAPEC and CKC to interpret learners' actions and strategies towards unleashing cyber-attacks. The assimilation of these standards is a determining factor in creating an organized and parameterized environment where learners' actions are monitored, recorded and dynamically assessed. 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 (presented in the segment 5 of table 7). In such way, the training and learning sessions created in HackLearn can be adapted to the participants' needs and capabilities.

Besides, the HackLearn's characteristics listed in the *Issues and Challenges* segment allow a preliminary estimation that HackLearn provides hands-on cyber security learning and training approaches with lower preparation and running costs compared to live competitions. Once a 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 scenarios

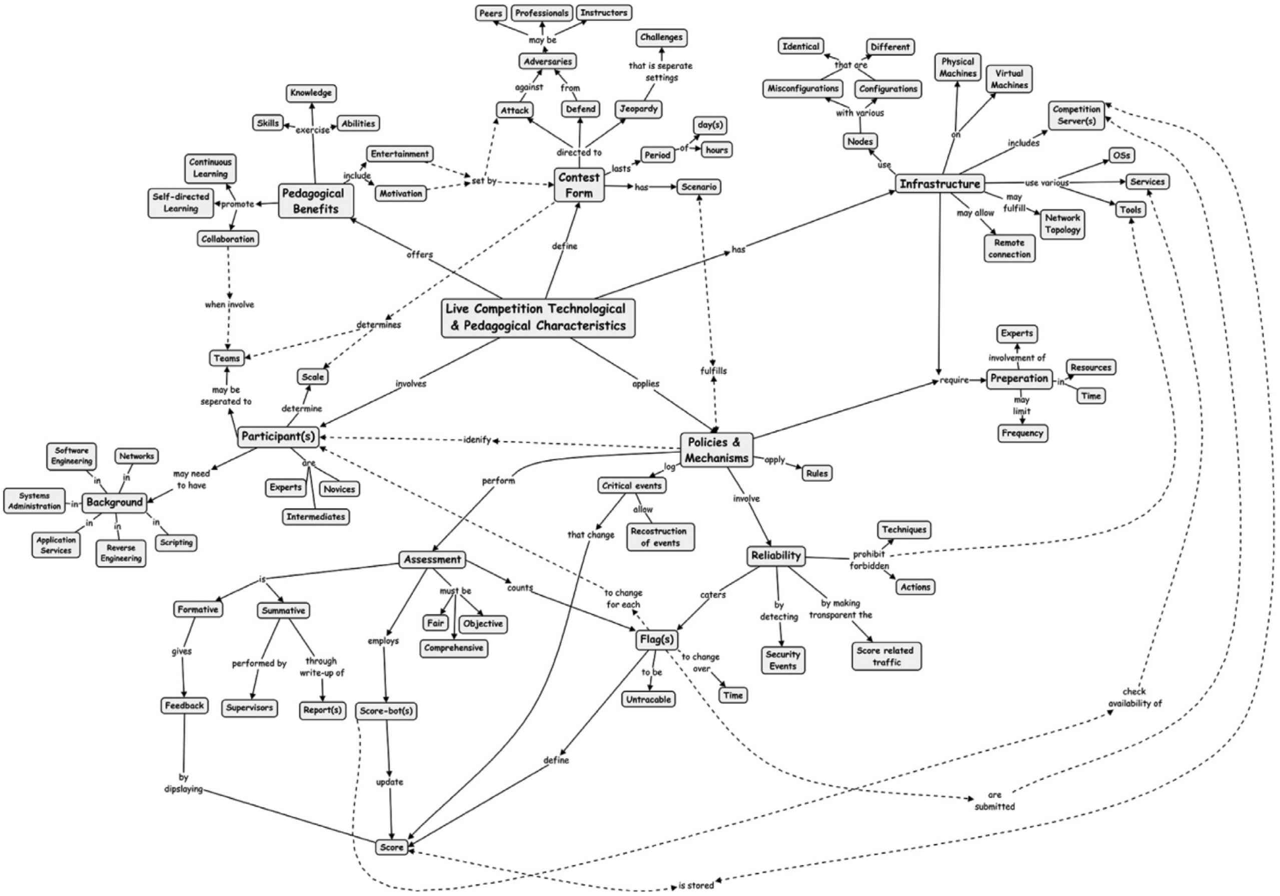

Fig. 11 – CtFs concept map (Katsantonis et al., 2017a).

includes a certain degree of logical complexity, the formation of scenarios is facilitated through the description of the scenarios' structural elements in the COFELET ontology and the reuse of objects stored in the repositories.

HackLearn has an 'always-on' architecture that allows learner to use it anytime and anywhere. Nevertheless, HackLearn is a game-based learning approach that is 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 development. However, in the first iteration of the study, the inclusion of the multiplayer feature was considered infeasible because it raises 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, learner mostly interacts with the game's terminal by entering 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 does.

8. Conclusion

The COFELET framework is an innovative multidisciplinary approach envisaging the development of effective cyber security game-based learning and training approaches. In this study, we examined how a cyber security serious game is structured, using the COFELET framework as a guide. The result of our study is the COFELET game life-cycle, a blueprint for developing COFELET games which presents the main components COFELET games contain and the manner the COFELET ontology elements are organized in the structure of such games. The COFELET game life-cycle was utilized for the design of HackLearn prototype COFELET game. Moreover, an excerpt of a multistep prototype scenario is presented to reveal practical design details.

To review our work, we evaluated HackLearn's design by employing an evaluation scheme based on key characteristics of cyber security game-based learning approaches. Since COFELET compliant approaches draw elements from CtF competitions, our evaluation scheme also considered relevant key characteristics of live competitions. The results of our preliminary evaluation are encouraging, as HackLearn embraces most of the key characteristics identified in cyber security game-based learning approaches and it additionally draws many key features from live competitions domain. The presented evaluation shows that HackLearn can provide authentic experiences to the learners, while it is a well-organized educational environment with clear learning objectives, scaffolding capabilities and appropriate assessment. Additionally, the presented evaluation allows us to make the preliminary estimation that HackLearn is expected to have much lower costs than live competitions. That is because it is considered less expensive to create new scenarios for a scenario-based game than organizing periodic live competition events, especially if we take into account that scenarios are not designed from scratch due to the reuse of the key elements created in the *Game Foundations* sub-phase (presented in sub-section 4.2).

Nevertheless, the evaluation revealed the 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 they will be furtherly studied in the subsequent phases of the current study and they will be included in the descendant games of HackLearn. In short-term the plans of the presented study include the finalization of HackLearn's implementation and the testing of HackLearn in real-world settings. Subsequently, we plan to enrich the repositories with elements, and to utilize them in different games and approaches.

Declaration of Competing Interest

None.

REFERENCES

Allen PD, Straub KA. Using Games to Enrich Continuous Cyber Training. Johns Hopkins APL Technical Digest 2015;33(2).

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

Ausubel DP. The Acquisition and Retention of Knowledge: A Cognitive View. Dordrecht: Springer - science + business media; 2000.

Carvalho MB, Bellotti F, Berta R, De Gloria A, Sedano CI, Hauge JB, Hu J, Rauterberg M. An activity theory-based model for serious games analysis and conceptual design. Computers & education 2015;87:166–81. doi:10.1016/j.compedu.2015.03.023.

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

Dewey J. How We Think: A Restatement of the Relation of Reflective Thinking to the Educative Process. Boston, Massachusetts: D.C. 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.

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 DH, 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. In: 2017 IEEE Global Engineering Education Conference (EDUCON); 2017. p. 771–9. doi:10.1109/EDUCON.2017.7942934.

Katsantonis MN, Fouliras P, Mavridis I. Conceptualization of Game Based Approaches for Learning and Training on Cyber Security. Proceedings of the 21st Pan-Hellenic Conference on Informatics, 2017.

Katsantonis M, Mavridis I. Ontology-Based Modelling for Cyber Security E-Learning and Training, 11841. Cham: Springer; 2019. Lecture Notes in Computer Science. doi:101007/978-3-030-35758-0_2.

Katsantonis MN, Kotini I, Fouliras P, Mavridis I. Conceptual Framework for Developing Cyber Security Serious Games. In: 2019 IEEE Global Engineering Education Conference (EDUCON), 2019. Dubai, United Arab Emirates,: IEEE; 2019. p. 872–81. https://ieeexplore.ieee.org/document/8725061/.

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

Mostafa M, Faragallah OS. Development of Serious Games for Teaching Information Security Courses. IEEE Access 2019;7:169293–305.

Mayer I. Towards a comprehensive methodology for the research and evaluation of serious games. Procedia Computer Science 2012;15:233–47.

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

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

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

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

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

Vygotsky LS. Mind in society: The development of higher psychological processes. Cambridge, Massachusetts: Harvard University Press; 1978.

MENELAOS KATSANTONIS is a *PhD Researcher* (Dept. of Applied Informatics, University of Macedo-nia, Greece). He holds a BSc (Computer Science, Uni-versity of Reading, UK) and an MSc (Distributed Sys-tems and Networks, University of Kent at Canterbury, UK). His current research interests include cybersecu-rity game-based learning and training. Since 2004, he has been serving, in an experienced *Informatics Instruc-tor's* capacity, various institutes and schools.

IOANNIS MAVRIDIS is a *Professor of Information Security* (Dept. of Applied Informatics, University of Macedonia, Greece). He also serves as *Director* of the Multimedia, Security & Networking (MSN) Lab. He holds a Diploma in Computer Engineering and Informatics from the University of Patras, Greece, and a PhD in Information Systems Security from the Aris-totle University of Thessaloniki, Greece. He serves as *Area Editor* (Cybersecurity) of *Array* (Elsevier). He has published more than 100 articles in journals and conferences. He is the author and co-author of 3 books on information se-curity. He has participated as principal investigator and researcher in several international and national funded R&D projects. His current research inte-rests include cybersecurity education, risk management, access control, cyber threat intelligence, digital forensics, and security economics.

DIMITRIS GRITZALIS is a *Professor of Cybersecurity* (Dept. of Informatics, Athens University of E-conomics & Business (AUEB), Greece). He also ser-ves as *Director* of the MSc Programme in Information Systems Development and Security, and *Director* of the INFOSEC Lab. He received a BSc (Mathematics, University of Patras, GR), an MSc (Computer Scien-ce, City University of New York, USA), and a PhD (Information Systems Security, University of the Ae-gean, GR). He serves as Academic Editor of *Compu-ters & Security* (Elsevier) and as Scientific Editor of the *International Jour-nal of Critical Infrastructure Protection* (Elsevier). He has served as *Asso-ciate Rector* for Research and *President* of the Lifelong Education Center of AUEB. He has also served as *President* of the Greek Computer Society and as *Associate Data Protection Commissioner* of Greece. He has publis-hed in several journals and conferences. His current research interests inc-lude cybersecurity education, critical infrastructure protection and malware.