Development of Ontology-Based Software Security Learning System with Contextualized Learning Approach

Shao-Fang Wen and Basel Katt
Norwegian University of Science and Technology, Gjøvik, Norway
Email: {shao-fang.wen, basel.katt}@ntnu.no

***Abstract*—Learning software security is one of the most challenging tasks in the information technology sector due to the vast amount of security knowledge and the difficulties in understanding the practical applications. The traditional teaching and learning materials, which are usually organized topically and security-centric, have fewer linkages with learners' experience and prior knowledge that they bring to the learning sessions. Learners often do not associate vulnerabilities or coding practices with programs similar to what they were writing in their previous time. Consequently, their motivation for learning is not touched by conventional methods. Therefore, it is necessary to develop learning tools that can improve learner' ability of application-scenarios connections by using a meaningful learning approach. In this paper, we present a software-security learning system based on ontologies that facilitates the contextual learning process by providing contextualized access to security knowledge via real software application scenarios, in which learners can explore and relate the security knowledge to the context they are already familiar with.**

***Index Terms*—software security, ontology, contextualized learning, learning system**

I. Introduction

Software security has been a subject of plethora studies for at least 40 years, and a steady stream of innovations has improved software engineers' ability to secure software development and to protect applications. Improving software security requires many different approaches. One way is to give software engineers or learners the knowledge and skills to resist attacks and handle errors appropriately [1]. To emphasize security, a relatively large number of best practices and vulnerability information have been published by security committees in publications or on the internet. To this extent, the huge amount of information has resulted in a form of information overload to learners. Moreover, the domain of software security is quite context-specific and can be applied in diverse ways [2]. As a result, learning software security becomes a complex and difficult task because learners must not only deal with a vast amount of knowledge about a variety of concepts and methods but also need to demonstrate the applicability of the knowledge through experience in order to understand their practical use.

In traditional software security teaching, little attention is given to what a real-world situation really means to learners, and there is not much content addressing the connection between the security concepts and learner' prior knowledge. In conventional security learning materials, the knowledge content is commonly security-centric and organized topically, which distinguishes two fundamental segments: the white-hat approach, where the main emphasis on security principles and anti-attack mechanisms, and the black-hat, which teaches how to break software and how malicious hackers write exploits. These learning materials are often described in the form of a reference manual or a guide to particular security subjects. The topical knowledge organization is useful for rote memorization of a specific security subject or for information reference later; however, it is difficult for learners to understand the rationale of the topics, and correlate those topics with real software scenarios. Learners usually finish reading such materials with little understanding of the context in which the security knowledge should be applied, or with the feeling that security domain is so extensive and software security is so difficult to achieve that they simply cast it aside.

We argue that the way learners process security information and their motivation for learning are not touched by conventional methods. Research indicates that learning is most efficient when it is linked with experience and prior knowledge that students bring to a given learning situation [3], [4]. However, novice learners do not always make connections between new information and prior knowledge or everyday experiences in ways that are productive for learning [5]. In the context of software security learning, learners interpret security knowledge they gain with a range of strongly held personal programming experience. They often do not associate vulnerabilities with programs similar to what they were writing in their previous time. As the suggestion given in the research of engineering education [6], establishing the relevance of learning materials before going into the details can provide the concrete experience that starts the learning process. In order to regulate learning about software security effectively,

Manuscript received March 21, 2019; revised July 25, 2019.

doi: 10.12720/jait.10.3.81-90

security knowledge should be contextualized in a meaningful scenario where they can learn security principles and processes with a real-world situation.

Our primary objective is to create conditions for more effective learning for software security that can motivate learners and stimulate their interests. This paper is part of an investigation into contextualized learning in the domain of software security. We propose a learning system, which facilitates the contextual learning process by providing contextualized access to security knowledge through real software application scenarios. This learning system is a place where learners can explore and relate the security knowledge to the context they are already familiar with. To develop this kind of learning system, the security knowledge should be modeled and managed in a manner where the knowledge can be retrieved taking the context of the application in hand into consideration. Ontologies make it possible to give this kind of purpose since it facilitates capture and construction of domain knowledge and enables representation of skeletal knowledge to facilitate the integration of knowledge bases irrespective of the heterogeneity of knowledge sources [7]. This paper presents the proposed design approach of the contextualized learning system and the developed proof-of-concept prototype.

The rest of this paper is organized as follows: In section 2, we introduce the theoretical background of this study. Section 3 reviews the related works on ontology approaches in the software security domain. In section 4, we describe the design approach of the contextualized learning system. Section 5 presents the detailed design of the underlying ontology of the learning system. Section 6 describes the developed prototype using the proposed approach. Lastly, the conclusion and future works are presented in section 7.

II. Theoretical Background

The theoretical background of this research is drawn from the field of context-based knowledge and contextualized learning. According to Anind K. Dey [8], context is "A set of information used to characterize a situation of an entity". Nonaka [9] indicates that knowledge reflects a particular stance, perspective, or intention in accordance with the characteristics of a specific context, which is different from information. According to Brézillon [10], [11], knowledge comes from a variety of context and it cannot be accurately understood without context. Without proper contextual information, knowledge can be isolated from other relevant knowledge resulting in limited or distorted understanding [12]. Researchers of psychology and education indicate when knowledge is learned in a context similar to that in which the skills will actually be needed, the application of learning to the new context may be more likely [8], [13], [14]. Predmore [15] shows that learning about knowledge content within real-world experience is important because "once [students] can see the real-world relevance of what they're learning, they become interested and motivated". Since context can give guidance about when, where and why a piece of knowledge is used, considering the context in knowledge use is very necessary to enhance the applicability of knowledge [1].

Contextualized teaching and learning builds upon a similar concept of putting learning activities into perspective to achieve the best teaching and learning outcomes. Researchers Berns and Erickson define contextualized learning as a practice that endeavors to link theoretical constructs that are taught during learning, to practical, real-world context [16]. The underlying theme behind contextual learning activities is simple. It recognizes that by embedding instructions in contexts that adult learners are familiar with, learners more readily understand and assimilate those instructions. Contextualized instruction in general, starts with presenting a context from which the concepts are developed on a need-to-know basis [17]. This requires teachers to teach in a more constructivist way, i.e. to position the concepts of the learning subject in contexts recognizable to students and to stimulate active learning of the students [18]. The contextualization of the learning on demand can not only be seen from the point of view of an actual problem or learning situation but also in a longer lasting process of learning activities that are integrated [19].

In computer science education, there is also a broad agreement that teaching units should start from a "real-world" context or phenomenon, aiming to create connections to prior knowledge, to increase the relevance of the material to students or to show application situations of the intended knowledge, thereby increasing motivation [20]-[23]. These contrast with more traditional approaches that cover abstract ideas first, before looking at practical applications. Likewise, in software engineering, studying from a context and then abstracting the knowledge gained to be able to use it in a new context is a common way of learning programming that has been observed extensively in both new and experienced programmers [24], [25]. In order to capture and use security knowledge appropriately, it is necessary to first specify which context information is to be handled, and then represent this in a format that is understandable and acceptable to the individuals. Thus, a context for a software security topic includes the circumstances in which its technical content exists. Therefore, to talk about software security in context is to say that knowledge would not only include the basic principles and processes of software security but would consider how security knowledge is used in one or more particular domains or application areas.

III. Related Work

In this section, we describe research works related to this study from the viewpoint of knowledge modeling support for software security based on ontology. According to Gruber [26], an ontology is "an explicit and formal specification of a conceptualization", that is, a formal description of the relevant concepts and relationships in an area of interest, simplifying and abstracting the view of the world for some purpose [27].

There have been a number of papers published in the area of ontology modeling and applying semantic technologies to software security. Some effort focused on building security ontology to model the security requirements. Salini and Kanmani [28] present an ontology of security requirements for web applications, including concepts of asset, vulnerabilities, threats, and stakeholders. Their work aims at enabling the reuse of knowledge about security requirements in the development of different web applications. Buch and Wirsing [29] present the SecWAO ontology with a focus on a secure web application, which aims to support web developers when specifying security requirements or making design decisions. It distinguishes concepts (classes) between methods, notations, tools, categories, assets, security properties, vulnerabilities, and threats.

Some research works present their ontology to support security design and risk assessment. Gyrard *et al.* [30] present the STACK ontology (Security Toolbox: Attacks & Countermeasures) to aid developers in the design of secure applications. STACK defines security concepts such as attacks, countermeasures, security properties, and their relationships. Countermeasures can be cryptographic concepts (encryption algorithm, key management, digital signature, and hash function), security tools, or security protocols. Kang and Liang [31] present a security ontology with the Model Driven Architecture (MDA) approach for the use in the software development process. The proposed ontology shows that the proposed security ontology can be used in modeling and designing security issues and concepts in each phase of the development process with MDA. Marques and Ralha [32] propose an ontology, which is related to the risk management aspect of web-based system development. The model is mainly employed in the design phase of the system development.

Finally, there are some papers focusing on using an ontology to model vulnerabilities and security attacks. Guo and Wang [33] present an ontology-based approach to model security vulnerabilities listed in Common Vulnerabilities and Exposures (CVE). The authors captured important concepts for describing vulnerabilities in the context of software security, providing machine-understandable CVE vulnerability knowledge and reusable security vulnerabilities interoperability. Khairkar *et al.* [34] present an ontology to detect attacks on web systems. The authors use semantic web concepts and ontologies to analyze security logs to identify potential security issues. This work aims to extract semantic relationships between attacks and intrusions in an Intrusion Detection System (IDS). Razzaq *et al.* [35] propose an ontology of attacks and an ontology of communication protocols, which provide a construct to improve the detection capability of application-level attacks in web application security. The authors employ the use of semantics in application layer security contrary to tradition signature-based approaches.

IV. Design Approach

To facilitate contextualized learning about software security and create engaging learning experiences for learners, we proposed a contextualized approach for software-security learning with three strategies: (1) Starting with a meaningful scenario; (2) Stimulating learners' mental model for software security learning; and (3) Moving from concrete to abstract security knowledge. Fig. 1 depicts an abstract representation of our design approach to the learning system for software security. Learners will engage in the learning process by taking advantage of relevant knowledge content. We describe in details these strategies in the following sections.

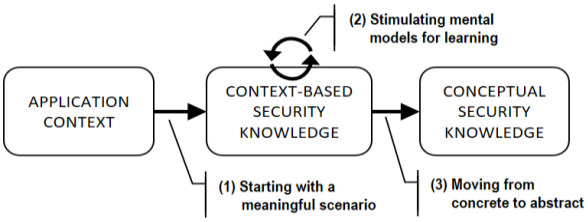

Figure 1. The design approach of the learning system

A. Starting with a Meaningful Scenario

Contextualized learning often takes the form of real-world examples or problems that are meaningful to the learners personally [36]. Creating the relevance of the learning knowledge before going into the details could provide a stronger foundation for the learning process. Therefore, to begin the process of learning, a meaningful situation for learners must first be established. In our study, the learning situations are created through the use of contextual scenarios in the application context, which utilize some form of anchoring situation events [37] to engage learners with security concepts that are addressed in the software problem or situation. Contextual scenarios refer to different manifestations within a context [38]. We choose a scenario-based approach because scenarios can be easily adapted to the situation of the represented applications and can be easily integrated with the conceptual security knowledge.

An anchoring event (i.e., the scenario in our study), enabling learners to visualize how the knowledge substance relates to their prior experience [37], could be revisited repeatedly during the learning sessions. For instance, regarding the application functionality of "Generating HTML pages" in web application context there includes a set of scenarios, such as generating static or dynamic pages, and using external data from HTTP requests or data stores. Those scenarios can serve as anchoring events to evoke the learners' memories of programming and draw attention to software events and conditions. Research has shown that using anchoring events in learning promotes memory recall and the subsequent transfer of information to a new setting [37], meanwhile, helps render abstract ideas more concretely and thus provides a cognitive mooring around which newly learned ideas can be linked with learners' prior understandings [39], [40]. The use of anchor evens in our study aims to echo learners' real-world experiences to context-based security knowledge to help learners apply their emerging understandings about software security to the real software cases, thus helping them see value in their learning sessions.

B. Stimulating Mental Models for Learning

Contextual learning is a learning approach that ties brain actions in creating patterns that have meaning [41]. In order to help learners make sense of complex security knowledge and create a strong and lasting bond among security concepts while they are engaged through various anchoring events, our strategy is to elicit learners' mental models for the navigation of security knowledge. Kenneth Craik [42] suggested that the human mind builds and constructs "small-scale models" to anticipate events. Such mental models allow learners to gain insight regarding their world by building a work scheme [43], which makes it easier for them to access the information needed to understand the knowledge domain, make predictions, and decide upon action to take [44]. This can result in successful learning by engaging students, fostering their concentration, and assisting them in organizing systemic information [45].

Mental models combine a schema or a knowledge structure with a process for manipulating the information in the memory [46], where the knowledge structure interrelates a collection of facts or concepts about a particular topic [47]. In order to be useful explanatorily, a mental model has to have a similar relation-structure to the reality it models. Then the constructed mental model can be used to answer questions or solve problems [48]. Generally, our intention was to guide learners in answering three questions while dealing with each anchoring event:

- What are the possible attacks?
- Why does it encounter attacks?
- How can these attacks be prevented?

The knowledge structure serves as the basis for both knowledge retention and retrieval, as well as transfer. Once learners answer what–why–how questions, the relationships between the security concepts are revealed in their midst, and thus, their representation of mental models expands.

C. Moving from Concrete to Abstract Knowledge

To help learners gain a more flexible understanding of the study concept in a range of situations with varying levels of abstraction, we organize security knowledge by blending abstract and concrete perspectives; presenting it with a sequence from concrete to abstract. In our study, abstract knowledge refers to the conceptual security domain knowledge while concrete knowledge relates to the contextualized scenario-specific security knowledge. Research has shown that presenting knowledge in both concrete and abstract terms are far more powerful than presenting either one in isolation [49]. Lave and Wenger [50] also argued that abstract and generalized knowledge gains its power through the expert's ability to apply it in specific situations. The used concrete-to-abstract approach in knowledge presentation differs from the traditional, where the concepts are of foremost importance and are usually explained first before concrete examples and applications are discussed. Consequently, learners may struggle to finish reading them due to a learning style mismatch. Several studies [51]-[53] have shown that the majority of engineering students are sensor-type learners, who like facts, data, and observable phenomena as opposed to theoretical abstractions. Deductive reasoning is facilitated when the domain is familiar and concrete rather than abstract [54].

In such concrete-to-abstract knowledge presentation, learners discover meaningful relationships between practical functions and abstract knowledge in the context of real applications. The value of concrete representations has been frequently noted in education. Concrete materials can support abstract reasoning because they can be explicitly designed to promote true inferences from perceptual representations to abstract principles [55]. A method known as concreteness fading [56] has the advantage of initially presenting concepts in a concrete fashion and then, over time, augmenting that initial presentation with progressively more abstract representations of the concepts. Abstract understanding is most effectively achieved through experience with perceptually rich, concrete representations [57], while concrete materials make concepts real and therefore easily internalized [58]. As long as the concrete knowledge and the underlying abstract explanation are understood by learners, learning transfers from one context to another will be more effective.

V. Underlying Ontology-Based Knowledge Model

One of the central ideas embedded within the learning system is to develop a kernel ontology-based security knowledge model. With this model, the learning application can handle contextualized security knowledge with multiple scenarios in different application-specific contexts and integrates security concepts of security domain knowledge.

A. Application Context Modeling

The context model represents a definition of what context is in a specific domain. In our ontology, the context for software security knowledge is supported by the creation of scenarios in different application contexts. The scenario presents a snapshot of possible features and corresponding code fragments in the specific functionality that is included in the Instruction class. It also draws on situated security knowledge, that is, understandings particular to the application context in which they generate. Fig. 2 represents the application context model used in the ontology. In the context modeling, in addition to scenarios, we focus on characteristics that are highly relevant for retrieval within a software application, concerning three perspectives:

- The application category that scenario/functionality belongs to,
- The platforms that the scenario functionality used, and
- The functional area (and the corresponding functionalities) that the application associated with.

Application category: It is a set of characteristics to categorize software applications, which include two sub-classes: paradigms (e.g., web, mobile, and desktop applications etc.) and the domains (e.g., banking, health, and logistics applications etc.).

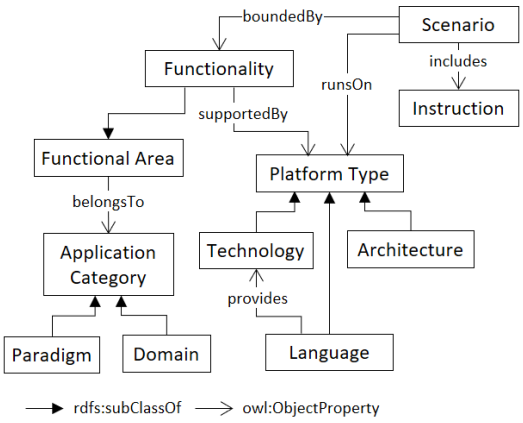

Figure 2. Application context model

Platform type: This superclass specifies programming languages, technologies, and architectures that are used to create the software application. Technology can be provided by a certain programming language. For example, Silverlight is the technology that has been implemented in C# language, while J2EE is the subset of Java technologies. Architectures refer to the fundamental system structure to operate the application, such as the MySQL database management system and the Android operating system.

Functional area: It is a group of application functionalities, which represents an aspect of software applications that can be performed by users or other systems in a particular application category. For example, outputting HTML is a functional area in the web-application paradigm, in which generating HTML dynamically using user-supplied data is one the functionalities. A functionality is supported and run on some combinations of platform types.

B. Security Domain Modeling

The security domain model describes the knowledge that is an object of teaching through a set of concepts (topics to be taught). In this model, we aim to design a security knowledge structure (schema) that is easier to store in the learners' memory for learning. For the purpose, the schema should be simplified and kept to the point for reducing the content load. We, therefore, identified three security concepts that are most widely used throughout the security domain and need to be concentrated learning on. Ultimately, three classes were incorporated into the security domain model: *Security Attack*, *Security Weakness*, and *Security Practice*. The definitions of the three security concepts are given in the following

Security attack: It represents actions taken against the software application with the intention of doing harm. Examples are SQL injection, Cross-Site Scripting, etc. Security attacks exploit security weakness existed in software applications.

Security practice: It represents methods, procedures or techniques to prevent security weakness.

Security weakness: It represents bug, flaws, vulnerabilities and other errors exist in the software applications.

From a security conceptualization point of view, we only want to indicate which principles or abstract ideas are needed, not their practical implementation. Therefore, we describe security knowledge in this model at a level of abstraction. The instances of these classes specify only the fundamental characteristics of the security concepts, not specific software application aspects. The main advantage of this design is to share a common understanding of the conceptual security knowledge among different security contexts. Furthermore, we adopt an abstract class *Security Domain* as a superclass for all security concepts. In the security domain model, we apply separation of concerns so that only very general descriptions remain as attributes in the class *Security Domain*. Additionally, for convenience, we allow grouping domain knowledge in categories, which themselves can belong to security concepts. Fig. 3 illustrates the security concepts and their relationships in the security domain model.

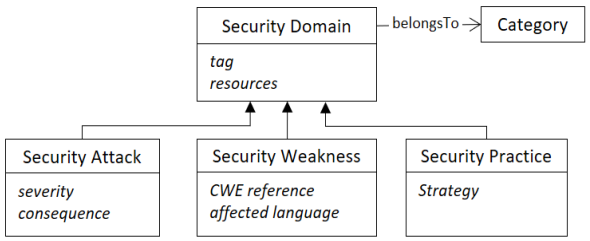

Figure 3. Security domain model

C. Security Contextualization Modeling

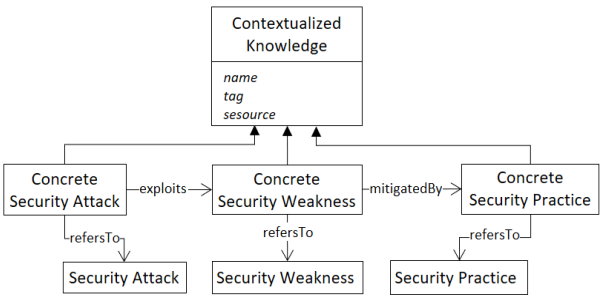

Figure 4. Security contextualization model

Fig. 4 illustrates the security contextualization modeling. The term contextualization is used here to describe the process of drawing specific connections between security domain knowledge being taught and an application context in which the conceptual knowledge can be relevantly applied or illustrated. To this extent, the security contextualization modeling manages security knowledge in the context of specific scenarios and brings together the conceptual knowledge that is described in the security domain model. The including security concepts are aligned with those defined in the security domain model, which are *Security Attack*, *Security Weakness*, and *Security Practice*. However, in order to clearly state the purposes and distinguish them from the security domain model, we use different classes, namely *Concrete Security Attack*, *Concrete Security Weakness*, and *Concrete Security Practice*. The abstract class *Contextualized Knowledge* is used from which these three

classes inherit common attributes such as tags or external resources. Once the conceptualization knowledge model is defined, each security concept is able to be connected to the corresponding classes in the security domain model. Fig. 5 shows the completed ontology-based knowledge model including the interrelationships of the components.

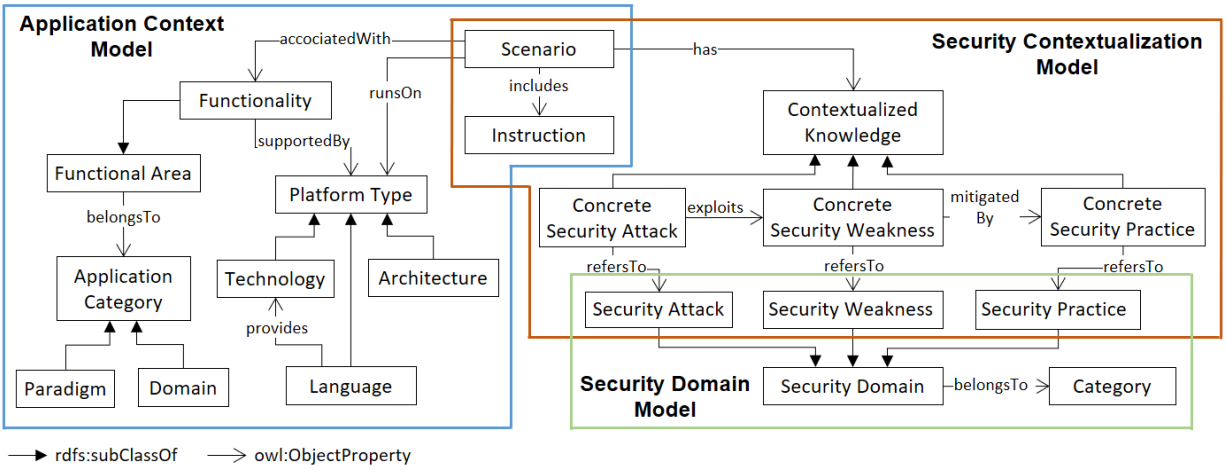

Figure 5. The ontology-based security knowledge model

VI. The Developed Prototype

We have developed a proof-of-concept prototype to demonstrate the proposed design approach. The high-level system architecture diagram is presented in Fig. 6. The front-end was designed as a web-based user interface with PHP and JavaScript languages and through it, learners can access the knowledge content. The backend was implemented in Java and access to the ontology repository was provided through the Jena API[1], a Java framework for building semantic web applications. Jena provides extensive Java libraries for helping developers develop code that handles RDF, OWL, and SPARQL in line with published W3C recommendations[2].

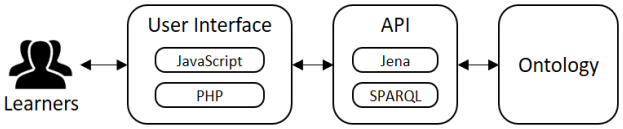

Figure 6. High-level system architecture diagram

A. Construction of the Ontology

To construct the ontology, we used Protégé and OWL[3] Editor because of its simplicity and popularity [59]. When searching the ontology, we use SPARQL protocol to extract information from the RDF graph. Fig. 7 depicts the ontology design in Protégé editor. An example of SPARQL and the executed result is presented in Fig. 8. The objective of this query is to return the instances of contextualized security knowledge of a specific scenario, and the short names of related security domain knowledge.

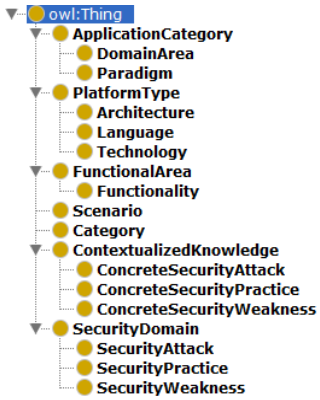

Figure 7. Ontology design in Protégé editor

B. The Process of Learning

The user interface of the prototyped system is presented in Fig. 9, in which a scenario of HTML output under the web application paradigm is demonstrated. In this prototype, the learning process begins with the concrete in a context familiar to learners and then gradually leads to an understanding of the abstract. Fig. 10 depicts the learning process that is constructed by the proposed learning system. First of all, a meaningful situation for learners must first be established. The access to learning contents in the learning application mainly happens scenario-oriented. We use the scenario as the starting point for learning security concepts on a need-to-know basis while presenting the modeled security knowledge. Based on the desired knowledge the learner selects relevant criteria from the application-context menu to scope the learning scenario. The instructional part of the scenario is made up of practical demonstrations of the pre-described application functionality and the code fragments behind it that bridge the corresponding security knowledge. As described previously, the selected scenario served as an anchoring event that can be view throughout the learning session to anchor learning in the learners' personal experience.

[1] https://jena.apache.org/
[2] https://www.w3.org/2001/sw/
[3] Web Ontology Language (OWL), a markup language based on Resource Description Framework/Extensible Markup Language (RDF/XML).

SPARQL query:

```
PREFIX onto: <http://folk.ntnu.no/shaofanfw/ContextOntology.owl#>
SELECT ?ConcreteSecurityAttack ?ConcreteSecurityWeakness ?ConcreteSecurityPractice
        ?attack_name ?weakness_name ?practice_name
            WHERE {
                        ?Scenario onto:has ?ConcreteSecurityAttack.
                        ?ConcreteSecurityAttack onto:exploits ?ConcreteSecurityWeakness.
                        ?ConcreteSecurityWeakness onto:mitigatedBy ?ConcreteSecurityPractice.
                        ?ConcreteSecurityAttack onto:relatesTo ?SecurityAttack.
                        ?SecurityAttack onto:attack_name ?attack_name.
                        ?ConcreteSecurityWeakness onto:relatesTo ?SecurityWeakness.
                        ?SecurityWeakness onto:weakness_name ?weakness_name.
                        ?ConcreteSecurityPractice onto:relatesTo ?SecurityPractice.
                        ?SecurityPractice onto:practice_name ?practice_name.
                        FILTER regex(str(?Scenario),'Scenario2', 'i')
}
```

ConcreteSecurityAttack	ConcreteSecurityWeakness	ConcreteSecurityPractice	attack_name	weakness_name	practice_name
SA-C-001	SW-C-001	SP-C-001	"Stored Cross-Site Scripting"	"Improper Neutralization of input"	"Input validation"
SA-C-001	SW-C-002	SP-C-002	"Stored Cross-Site Scripting"	"Failure to escape/encode ouput"	"Output encoding

Figure 8. An example of SPARQL and the executed result

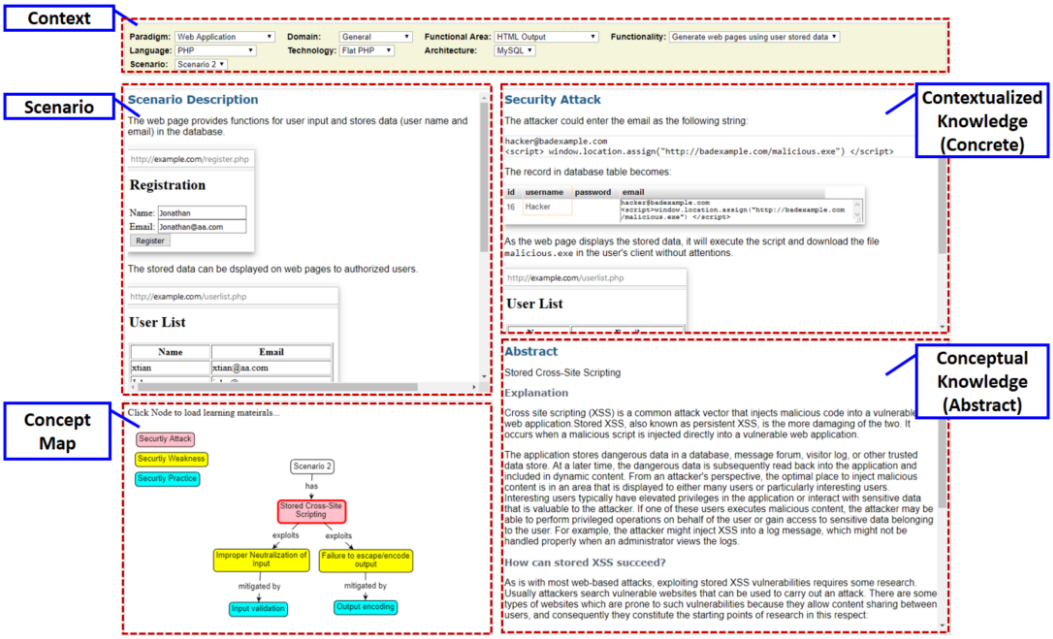

Figure 9. The user interface of the developed prototype

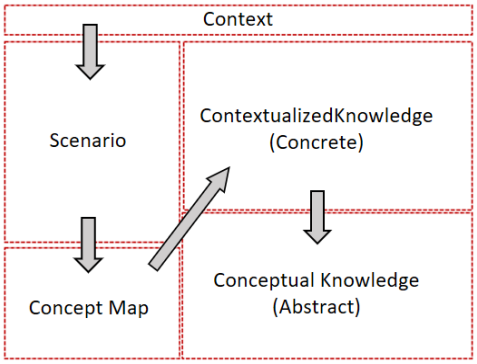

Figure 10. The constructed learning process of the learning system

To guide learners navigating through the contextualized knowledge efficiently, it is necessary to illustrate the relationship between the security concepts. On the one hand, it must be transparent for learner about, which causes and effects relevant to the learning content he (or she) is studying. On the other hand, this is essential for learners in order to integrate the semantical impact of the knowledge structure into the mental models for efficient learning. For the purpose, we outline the learning contents in a graphical *Concept Map*, which is shown in the left corner of the system appearance. Concept Map is a visual representation of different concepts and their relationships. Concept mapping help in organizing learners' knowledge by integrating information into a progressively more complex conceptual framework [60]. With the use of concept mapping, the learning arena can be virtualized in a learner's mind [61]. From the visual description, learners extract propositions and create a mental model from the graph. Meanwhile, the extracted mental model will be inherently influenced by connecting to their prior experience.

The design of our ontology is able to provide the basis for the development of the concept map of the relationship between these concepts. While a node is clicked on the concept map, the relevant knowledge content is displayed in the right half of the appearance, where the upper part is the contextualized knowledge and the lower part is an abstract explanation, following the concrete-to-abstract presentation strategy. By concrete representations, we include perceptually detailed and rich materials, such as demonstrating security attacks with different exploits, identifying mistakes in the source code, and showing the secure coding practices to fix the mistakes. Fig. 11 shows a system appearance of viewing security weakness of the scenario. With scenario-

description presenting aside, learners can easily recall features of the context (e.g. code fragment) without interrupting the learning process. After experiencing the facts, learners then move on to conceptual knowledge, where the abstract explanation is presented. Therefore, dynamic, e.g., situational application scenario is integrated together with the conceptual security domain knowledge. Fig. 12 presents another scenario in the paradigm of "General implementation" and the language of C/C++. This demonstrated scenario introduces security knowledge related to the functionality of "Performing memory buffer operations using user-supplied data".

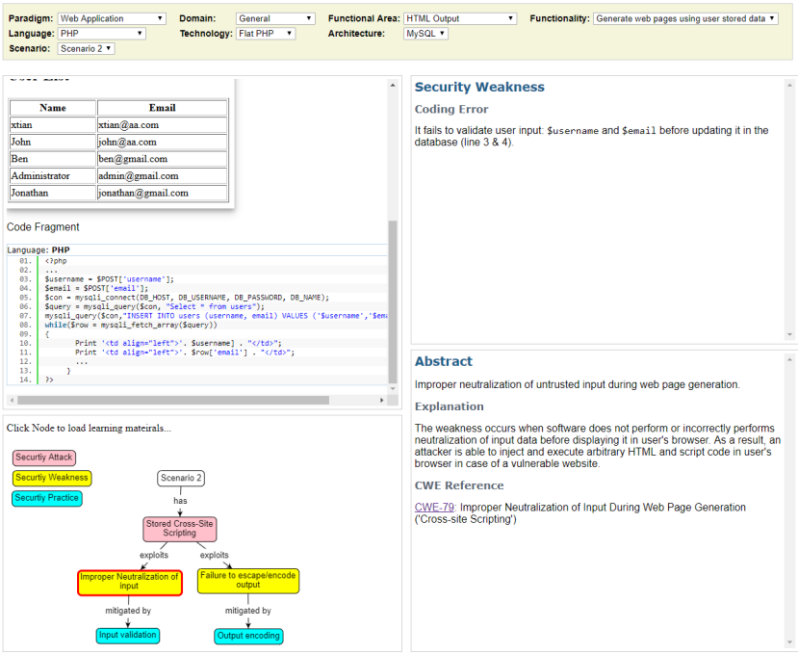

Figure 11. The screen shot of viewing security weakness of the scenario

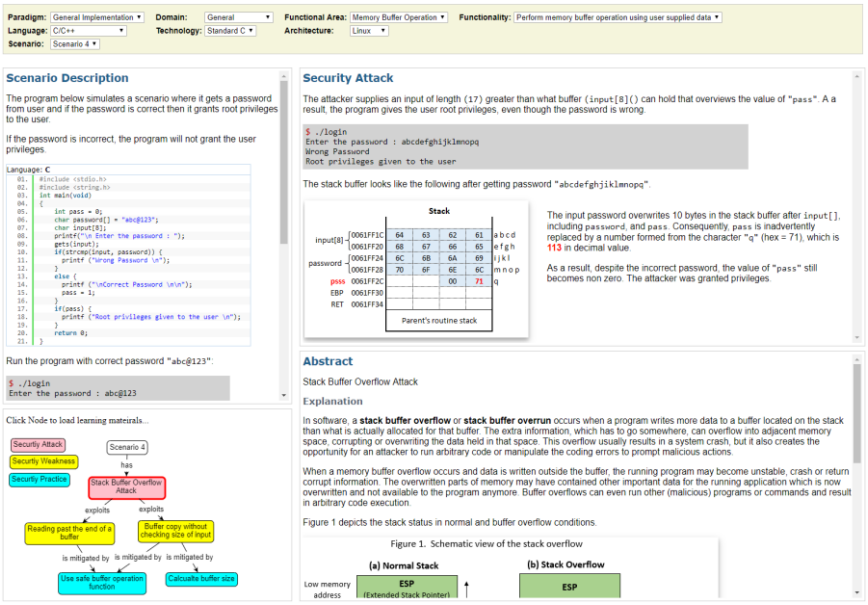

Figure 12. A scenario for memory buffer operations in C/C++

VII. Conclusion and Future Work

This paper presents an ontology-based learning system for software security learning with a contextualized learning approach, which contains three strategies. The first is to establish meaningful scenarios to create a meaningful situation for learners. The design of the application context aims to activate the learner's prior knowledge of software programming and anchors the learning about security knowledge. The second strategy is to organize underlying security knowledge in a structured manner that can stimulate learners' mental models to support more efficient learning in the specified context. The third is to guide learners to engage with concrete knowledge before studying abstract knowledge. This strategy assists learners in discovering meaningful concepts and relationships between practical functions and abstract knowledge when working in this context.

Our research attempts to place security learning in the context of real application scenarios. The benefits of this contextualized approach can also be explained by the effective mechanism of intrinsic motivation, where a learner is drawn to engage in a task because it is perceived as interesting, enjoyable, and/or useful [62]-[64]. Since the given context is connected and relevant to their prior knowledge and life experiences in software development, security learning can then be related to a similar programming topic that they want to learn about or a problem to be solved. We strongly believe this implies a direct effect of the contextualized learning approach on higher overall learning satisfaction, which motivates students to learn.

Our future work includes improving the usability of the user interface and enriching the knowledge content with a variety of application scenarios. We plan as well as learning experiments with bachelor students, in order to validate our proposal.

REFERENCES

[1] M. Bishop, "A clinic for "Secure" programming," *IEEE Security & Privacy*, vol. 8, no. 2, 2010.

[2] G. McGraw, "Software security: Building security in," in *Proc. 17th International Symposium on Software Reliability Engineering*, 2006.

[3] N. R. Council, *How People Learn: Brain, Mind, Experience, and School: Expanded Edition*, National Academies Press, 2000.

[4] J. Leach and P. Scott, "Individual and sociocultural views of learning in science education," *Science & Education*, vol. 12, no. 1, pp. 91-113, 2003.

[5] S. M. Land, "Cognitive requirements for learning with open-ended learning environments," *Educational Technology Research and Development*, vol. 48, no. 3, pp. 61-78, 2000.

[6] R. M. Felder, *et al.*, "The future of engineering education II. Teaching methods that work," *Chemical Engineering Education*, vol. 34, no. 1, pp. 26-39, 2000.

[7] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?" *International Journal of Human-Computer Studies*, vol. 43, no. 5, pp. 907-928, 1995.

[8] A. K. Dey, Understanding and using context," *Personal Ubiquitous Computing*, vol. 5, no. 1, pp. 4-7, 2001.

[9] I. Nonaka and N. Konno, "The concept of "ba": Building a foundation for knowledge creation," *California Management Review*, vol. 40, no. 3, pp. 40-54, 1998.

[10] P. Brézillon, "Modeling and using context: Past, present and future," Rapport de recherche interne LIP6, Paris, 2002.

[11] P. Brézillon and J. C. Pomerol, "Contextual knowledge sharing and cooperation in intelligent assistant systems," *Le Travail Humain*, pp. 223-246, 1999.

[12] G. Goldkuhl and E. Braf, "Contextual knowledge analysis-understanding knowledge and its relations to action and communication," in *Proc. Second European Conference on Knowledge Management*, 2001.

[13] D. Perin, "Facilitating student learning through contextualization: A review of evidence," *Community College Review*, vol. 39, no. 3, pp. 268-295, 2011.

[14] D. H. Dolmans, *et al.*, "Problem-based learning: Future challenges for educational practice and research," *Medical Education*, vol. 39, no. 7, pp. 732-741, 2005.

[15] S. R. Predmore, "Putting it into context," *Techniques: Connecting Education and Careers*, vol. 80, no. 1, pp. 22-25, 2005.

[16] R. G. Berns and P. M. Erickson, "Contextual teaching and learning: Preparing students for the new economy," *Constructivism*, vol. 9, 2001.

[17] J. Bennett, F. Lubben, and S. Hogarth, "Bringing science to life: A synthesis of the research evidence on the effects of context-based and STS approaches to science teaching," vol. 91, no. 3, pp. 347-370, 2007.

[18] I. Parchmann, *et al.*, ""Chemie im Kontext": A symbiotic implementation of a context-based teaching and learning approach," *International Journal of Science Education*, vol. 28, no. 9, pp. 1041-1062, 2006.

[19] M. Specht, "Designing contextualized learning," in *Handbook on Information Technologies for Education and Training*, Springer, 2008, pp. 101-111.

[20] S. Cooper and S. Cunningham, "Teaching computer science in context," *ACM Inroads*, vol. 1, no. 1, pp. 5-8, 2010.

[21] M. Guzdial, "Does contextualized computing education help?" *ACM Inroads*, vol. 1, no. 4, pp. 4-6, 2010.

[22] I. Diethelm, P. Hubwieser, and R. Klaus, "Students, teachers and phenomena: Educational reconstruction for computer science education," in *Proc. the 12th Koli Calling International Conference on Computing Education Research*, 2012.

[23] M. Guzdial, "Teaching computing for everyone," *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 6-6, 2006.

[24] A. J. Ko and B. A. Myers, "Debugging reinvented: Asking and answering why and why not questions about program behavior," in *Proc. the 30th International Conference on Software Engineering*, 2008.

[25] A. Apvrille and M. Pourzandi, "Secure software development by example," *IEEE Security & Privacy*, vol. 3, no. 4, pp. 10-17, 2005.

[26] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.

[27] Y. Wand, V. C. Storey, and R. Weber, "An ontological analysis of the relationship construct in conceptual modeling," *ACM Transactions on Database Systems (TODS)*, vol. 24, no. 4, pp. 494-528, 1999.

[28] P. Salini and S. Kanmani, "Ontology-based representation of reusable security requirements for developing secure web applications," *International Journal of Internet Technology and Secured Transactions*, vol. 5, no. 1, pp. 63-83, 2013.

[29] M. Busch and M. Wirsing, "An ontology for secure web applications," *Int. J. Software and Informatics*, vol. 9, no. 2, pp. 233-258, 2015.

[30] A. Gyrard, C. Bonnet, and K. Boudaoud, "The stac (security toolbox: Attacks & countermeasures) ontology," in *Proc. the 22nd International Conference on World Wide Web*, 2013.

[31] W. Kang and Y. Liang, "A security ontology with MDA for software development," in *Proc. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2013.

[32] M. Marques and C. G. Ralha, "An ontological approach to mitigate risk in web applications," in *Proceedings of SBSeg*, 2014.

[33] M. Guo and J. A. Wang, "An ontology-based approach to model common vulnerabilities and exposures in information security," in *Proc. ASEE Southest Section Conference*, 2009.

[34] A. D. Khairkar, D. D. Kshirsagar, and S. Kumar, "Ontology for detection of web attacks," in *Proc. International Conference on Communication Systems and Network Technologies*, 2013.

[35] A. Razzaq, *et al.*, "Ontology for attack detection: An intelligent approach to web application security," *Computers & Security*, pp. 124-146, 2014.

[36] A. E. Rivet and J. Krajcik, "Contextualizing instruction: Leveraging students' prior knowledge and experiences to foster understanding of middle school science," *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, vol. 45, no. 1, pp. 79-100, 2008.

[37] Cognition Technology Group at Vanderbilt, "Anchored instruction in science and mathematics: Theoretical basis, developmental projects, and initial research findings," *Philosophy of Science, Cognitive Psychology*, pp. 244-273, 1992.

[38] E. P. Errington, "Being there: Closing the gap between learners sand contextual knowledge using near-world scenarios," *International Journal of Learning*, vol. 16, pp. 585-594, 2009.

[39] Cognition Technology Group at Vanderbilt, "The Jasper series as an example of anchored instruction: Theory, program description, and assessment data," *Educational Psychologist*, vol. 27, no. 3, pp. 291-315, 1992.

[40] R. D. Sherwood, *et al.*, "Some benefits of creating macro-contexts for science instruction: Initial findings," *Journal of Research in Science Teaching*, vol. 24, no. 5, pp. 417-435, 1987.

[41] R. Davtyan, "Contextual learning," in *Proc. ASEE 2014 Zo. 1 Conf.*, 2014.
[42] K. J. W. Craik, "The nature of explanation," *CUP Archive*, vol. 445, 1967.
[43] D. Gentner and A. L. Stevens, *Mental Models*, Psychology Press, 2014.
[44] W. B. Rouse and N. M. Morris, "On looking into the black box: Prospects and limits in the search for mental models," *Psychological Bulletin*, vol. 100, no. 3, p. 349, 1986.
[45] N. M. Seel, S. Al-Diban, and P. Blumschein, "Mental models & instructional planning," in *Integrated and Holistic Perspectives on Learning, Instruction and Technology*, Springer, 2000, pp. 129-158.
[46] M. D. Merrill, "Knowledge objects and mental models," in *Proc. International Workshop on Advanced Learning Technologies*, 2000.
[47] Psychology Wiki. [Online]. Available: http://psychology.wikia.com/wiki/Knowledge_structure
[48] D. E. Kieras and S. Bovair, "The role of a mental model in learning to operate a device," *Cognitive Science*, vol. 8, no. 3, pp. 255-273, 1984.
[49] H. Pashler, *et al.*, "Organizing instruction and study to improve student learning," *National Center for Education Research*, vol. 76, no. 10, 2007.
[50] J. Lave, E. Wenger, and E. Wenger, *Situated Learning: Legitimate Peripheral Participation*, Cambridge: Cambridge University Press, 1991.
[51] M. H. McCaulley, "Psychological types in engineering: Implications for teaching," *Engineering Education*, vol. 66, no. 7, pp. 729-736, 1976.
[52] M. H. McCaulley, *et al.*, "Applications of Psychological type in engineering-education," *Engineering Education*, vol. 73, no. 5, pp. 394-400, 1983.
[53] R. M. Felder and L. K. Silverman, "Learning and teaching styles in engineering education," *Engineering Education*, vol. 78, no. 7, pp. 674-681, 1988.
[54] P. C. Wason and D. Shapiro, "Natural and contrived experience in a reasoning problem," *The Quarterly Journal of Experimental Psychology*, vol. 23, no. 1, pp. 63-71, 1971.
[55] M. Bassok, "Using content to interpret structure: Effects on analogical transfer," *Current Directions in Psychological Science*, vol. 5, no. 2, pp. 54-58, 1996.
[56] R. L. Goldstone and Y. Sakamoto, "The transfer of abstract principles governing complex adaptive systems," *Cognitive Psychology*, vol. 46, no. 4, pp. 414-466, 2003.
[57] R. L. Goldstone and J. Y. Son, "The transfer of scientific principles using concrete and idealized simulations," *The Journal of the Learning Sciences*, vol. 14, no. 1, pp. 69-110, 2005.
[58] P. Kamina and N. N. Iyer. (2009). From concrete to abstract: Teaching for transfer of learning when using manipulatives. *NERA Conference Proceedings*. [Online]. Available: https://opencommons.uconn.edu/nera_2009/6
[59] T. Tudorache, *et al.*, "WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web," *Semantic Web*, vol. 4, no. 1, pp. 89-99, 2013.
[60] D. Ifenthaler and R. Hanewald, *Digital Knowledge Maps in Education: Technology-Enhanced Support for Teachers and Learners*, Springer Science & Business Media, 2013.
[61] N. Shambaugh, "The cognitive potentials of visual constructions," *Journal of Visual Literacy*, vol. 15, no. 1, pp. 7-24, 1995.
[62] C. A. Kozeracki, "Preparing faculty to meet the needs of developmental students," *New Directions for Community Colleges*, vol. 2005, no. 129, pp. 39-49, 2005.
[63] R. J. Dean and L. Dagostino, "Motivational factors affecting advanced literacy learning of community college students," *Community College Journal of Research Practice*, vol. 31, no. 2, pp. 149-161, 2007.
[64] D. I. Cordova and M. R. Lepper, "Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice," *Journal of Educational Psychology*, vol. 88, no. 4, p. 715, 1996.

Shao-Fang Wen is a Ph.D. student at Norwegian University of Science and Technology (NTNU), Department of Information Security and Communication Technology. He is also a member of Center for Cyber and Information Security (CCIS). His research fields are security education and knowledge management.

Basel Katt is an associate professor at Norwegian University of Science and Technology (NTNU), Department of Information Security and Communication Technology. Basel Katt received his M.Sc. degree in Computer Science from University of Essen-Duisburg, Germany, and his PhD degree in Computer Science from the University of Innsbruck, Austria. His research interest lies in the areas of security engineering, security testing, model driven security, security education, access control, and anomaly detection. He joined NTNU in 2015, and he is leading the Norwegian Cyber Range lab.