社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
KBSE2011-24(2011-09)

ソフトウェアセキュリティ学習環境の開発

櫨山淳雄† 清水啓人‡

† 東京学芸大学 〒184-8501 東京都小金井市貫井北町 4-1-1
‡ 情報セキュリティ大学院大学 〒221-0835 神奈川県横浜市神奈川区鶴屋町 2-14-1
E-mail: † hazeyama@u-gakugei.ac.jp, ‡ mgs111003@iisec.ac.jp

あらまし インターネット上でのサービスの増大に伴いソフトウェアセキュリティの重要性が認識されている。またソフトウェアセキュリティ技術を身につけた人材育成の重要性も認識されている。本論文ではソフトウェアセキュリティ学習プロセスとその学習プロセスに基づく学習環境を提案している。学習プロセスでは、既存のソフトウェア開発演習の成果物を入力とし、ソフトウェアセキュリティに関する情報（標準、方法論、ガイドライン、パターン等）を参照しながら、セキュリティを考慮した成果物を作成するというものである。学習環境はソフトウェア開発成果物の蓄積、ソフトウェアセキュリティに関する情報の蓄積とこれらの間の関連づけ、成果物作成に至った根拠情報の蓄積とソフトウェアセキュリティ開発演習成果物との関連づけ、ソフトウェアセキュリティ開発演習成果物へのレビューコメントの蓄積とソフトウェアセキュリティ開発演習成果物への関連づけを可能にしている。

キーワード ソフトウェアセキュリティ，学習環境，知識共有

Development of a Software Security Learning Environment

Atsuo HAZEYAMA† and Hiroto SHIMIZU‡

† Tokyo Gakugei University 4-1-1 Nukuikita-machi, Koganei-shi, Tokyo, 184-8501 Japan
‡ Institute of Information Security 2-14-1 Tsuruya-machi, Kanagawa-ku, Yokohama-shi, Kanagawa, 221-0835 Japan
E-mail: † hazeyama@u-gakugei.ac.jp, ‡ qxggt630@yahoo.co.jp

Abstract Importance for software security technologies has been recognized according to increase of services on the internet. It is also pointed out that to foster human resources who have knowledge and skills on software security technologies is important. This paper proposes a learning process for software security and a learning environment that supports the learning process. In the learning process, learners create artifacts of software security (we call them “software security artifacts”) from those (we call them “software engineering artifacts”) from a traditional software engineering course without dealing with software security by referring to the reference information for software security (standards, methodologies, guidelines, security patterns, and so on). The learning environment supports to store “software engineering artifacts,” “software security artifacts,” the reference information and its association, rationale and association with “software security artifacts,” and review comments and association with “software security artifacts.”

Keyword Software Security, Learning Environment, Knowledge Sharing

1. はじめに

インターネット上でのサービスは増大の一途を辿っている。今後も携帯端末や情報家電などの普及により、この傾向は顕著なものになっていくと予想される。それに伴い、コンピュータセキュリティの重要性が高まってきている。特に近年、多くのサービスがソフトウェアで実現され、その複雑さが増大しているため、ネットワークセキュリティ技術(アクセス制御や暗号技術など)のみでなくソフトウェアセキュリティ技術の重要性が認識されるようになってきた[6]。ソフトウェアセキュリティはソフトウェア開発過程全体でセキュリティを扱う[6]。すなわち単にネットワークセキュリティ技術をシステムに組み込むというのではなく、開発工程の各段階においてセキュリティに関する作りこみを行い、セキュアなソフトウェアを開発することを目指す。また、ソフトウェアセキュリティ技術を熟

知した人材が不足している現状から、ソフトウェアセキュリティ技術を身に付けた人材育成の重要性が指摘されている[6, 8]。

著者らはこれまで大学においてソフトウェア開発教育に取り組んできた[3]。オブジェクト指向技術に基づくWebアプリケーション開発を対象としている。しかし、ソフトウェアセキュリティを体系的に学習するには至っていない。

そこで、これまで取り組んできたソフトウェア開発教育の延長線上にソフトウェアセキュリティ教育を位置づけ、その学習環境を提案する。

2. 学習プロセス

本論文で提案する学習プロセスの概観を図1に示す。これまでに取り組んできたソフトウェア開発演習(ソフトウェアセキュリティを扱っていない)の成果物(文献[4]のシステムに蓄積された成果物であり、図1で「ソフトウェア開発演習成果物」と記している)を入力として、セキュリティを考慮したソフトウェア開発を行う(この開発の成果物を図1では「ソフトウェアセキュリティ開発演習成果物」と記している)。

システムの通常の機能を検討した後にセキュリティについて検討を行うという学習の流れは、例えばUnified Modeling Language (UML)のユースケース図に描かれる通常の機能であるユースケースに対して、悪意のあるアクター(ミスアクター)による悪意のある行為であるミスユースケースを抽出し、ミスユースケース図を作成するというSindreとOpdahlによる手法[11]に符合する。

ソフトウェアセキュリティに関して実務経験に乏しい学習者がソフトウェアセキュリティの学習を行うことは困難であり、何らかの支援が必要である。例えば要求分析工程に対して、セキュリティ要求分析方法論(例えばSQUARE[15])、セキュリティパターン(例えば[9])、標準(例えばCommon Criteria[14])、情報リソース(例えばCAPEC[13])が公開されており、これらを教材として活用しながら開発を進めることを想定している。

開発において作成した成果物とそれらを作成する際に利用した情報とを関連付け、設計根拠を積極的に残させるようにする。これにより、学習のふり返りを促す。さらに作成された成果物に対してレビューなどのフィードバックを与え、成果物の品質向上を図る。

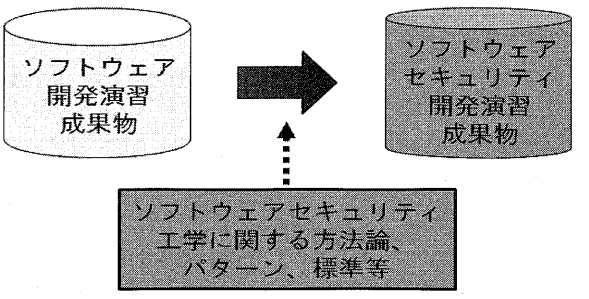

図1．提案する学習プロセス

3. 学習環境

3.1. 学習環境に対する要件

前節で提案した学習プロセスを支援する学習環境に対する要件を議論する。

要件1：ソフトウェアセキュリティ学習の入力となる成果物並びに学習の成果である成果物の登録・修正・削除等成果物管理ができること。

要件2：学習時に参照する情報(方法論、パターン、標準等。これらを参照情報と呼ぶ)の登録・修正・削除等とそれらの関係を記述できること。個々の参照情報を登録、参照できるとともに、個別の情報は相互に関係していることもある。そこで参照情報間の関係を設定するとともに、それらの関係の可視化ができること。

要件3:成果物作成に至った根拠となる情報(根拠情報)の登録・修正・削除ができること。そして、根拠情報と成果物、参照情報との関係を登録・修正・削除等ができること。BarnumとSethiはアーキテクチャ設計において使用した攻撃パターンをテスト工程において活用できるようにするため、文書化することの重要性を言及している[2]。

要件4：成果物に対するレビューコメントが付けられること。

3.2. 学習環境の設計

前項で述べた要件を満たす学習環境が提供する機能について述べる。図2はシステム全体の構成を示している。大きく4つの主要機能から構成されている。

- 図1の学習プロセスで示した2種類の成果物の登録・管理を行う成果物管理
- 参照情報(方法論、パターン、標準等)の構築
- 根拠の登録と関連する情報との関連の設定
- レビューコメントの付与

以下でそれぞれの詳細について説明する。

(1) 成果物管理

演習の入力となるソフトウェア開発演習成果物並びに演習の成果物であるソフトウェアセキュリティ開発演習成果物の登録が行えるようにする。

また、成果物の修正過程を追跡できるよう、成果物の版管理を行う。

さらにソフトウェアセキュリティ開発演習成果物にはいくつかの種類があるので、それらの間の関連を設定できるようにする。

(2) 参照情報(方法論、パターン、標準等)の構築

参照情報の登録・修正・閲覧・削除が行えるようにする。参照情報間には関連がある場合がある。そこで参照情報間の関連付けを行えるようにする。

(3) 成果物と参照情報・設計根拠の関連づけ、リンクによる相互参照

学習において学習者がどのような情報に基づき、どのように考え、成果物作成を行ったのかという根拠を記録するとともに、根拠とソフトウェアセキュリティ開発演習成果物との間に関連を設定できるようにする。

さらに、その根拠と関連する参照情報との関連を記録し、閲覧できるようにする。方法論を構成するステップと対応付けて成果物、参照情報、設計根拠を関連付けられるようにもする。

(4) ソフトウェアセキュリティ開発演習成果物に対するコメントづけ

ソフトウェアセキュリティ開発演習成果物に対してレビューコメントを記述でき、当該成果物との関連を設定できるようにする。このとき、レビューコメントの根拠となる参照情報との間にも関連を設定できるようにする。

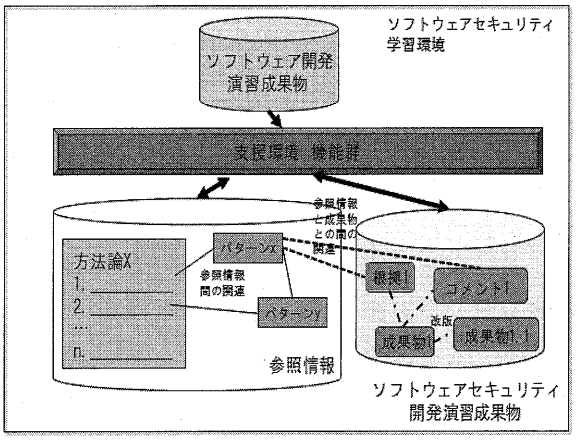

図 2　システムの構成

図 3 にシステム全体のエンティティタイプのみを対象としたクラス図を示す。以下で各クラスの役割、クラス間の関連について説明する。

クラス project はプロジェクトに関する情報を保持するクラスであり、プロジェクトの識別子(id)、プロジェクト名(name)、開始日(start_date)、終了日(end_date)を属性に持つ。

クラス artifact はソフトウェア開発演習成果物に関する情報を保持するクラスであり、成果物名(name)、成果物の種別(仕様書、ユースケース図等)を表すタイプ(type)、作成日(date)、ファイルへのパス(path)を属性に持つ。

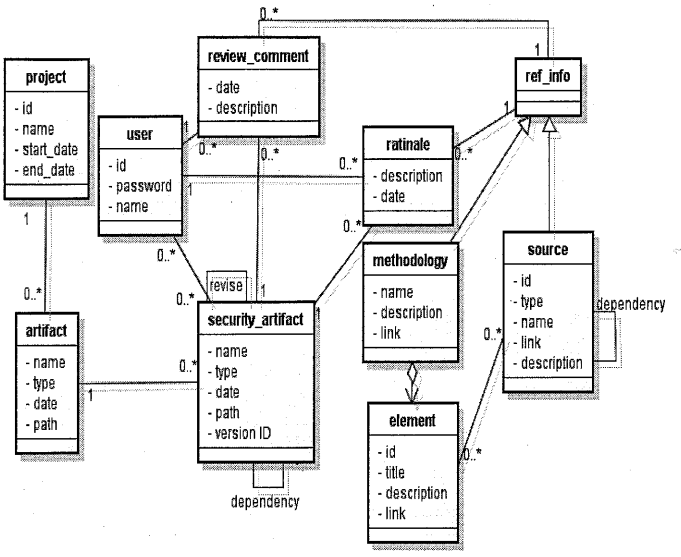

図 3　システムのクラス図

クラス security_artifact はソフトウェアセキュリティ開発演習成果物に関する情報を保持するクラスであり、成果物名(name)、成果物の種別(仕様書、ミスユースケース図等)を表すタイプ(type)、作成日(date)、ファイルへのパス(path)、版識別番号(versionID)を属性に持つ。

クラス ref_info は参照情報を表すクラスであり、方法論を表すクラス(methodology)とソース(source)クラスに大別される。methodology は方法論や標準の全体を表し、方法論名(name)、内容(description)、関連する情報へのリンク(link)を属性に持つ。方法論を構成する構成要素を element で表し、順序(id)、見出し(title)、内容(description)、関連する情報へのリンク(link)を属性に持つ。methodology はその要素であるクラス element と集約関係を有する。source は例えば 1 つのパターンを表し、識別子(id)、名前(name)、種別(パターン、ガイドライン等)を表すタイプ(type)、内容(description)、関連する情報へのリンク(link)を属性に持つ。

クラス rationale は根拠に関する情報を保持するクラスであり、根拠の内容(description) と根拠が記述された日付(date)を属性に持つ。

クラス review_comment はソフトウェアセキュリティ開発演習成果物に対するレビューコメントの情報を保持するクラスであり、コメントの内容(description)とコメントが記述された日付(date)を属性に持つ。

次に関連について説明する。

security_artifact は security_artifact と改版(revise)関連を有する。また、security_artifact 間には依存関係(dependency)関連を有する。これは、成果物間の追跡

可能性を表す。さらに、security_artifact は作成者である user との関連、その成果物の入力成果物である artifact との関連、作成者による作成に至った根拠を表す rationale との関連、security_artifact に対する他者からのレビューコメント review_comment との関連を有する。

source は methodology の要素である element と関連を有する。source のインスタンスは source の他のインスタンスと関連を有する。

rationale と review_comment はその作成者である user 並びにその根拠となる ref_info との関連を有する。

3.3. 試作

学習環境を Web アプリケーションとして試作した。

図 4 は方法論の詳細を表示した画面である。方法論を構成するステップとそれに関連づけられたソース情報の一覧が表示されている。

方法論名	Methodology by Sindre and Opdahl
概要	from eliciting security requirement with misuse case
リンク	http://www.idi.ntnu.no/~guttorm/
1	タイトル Asset identification 概要　アセットを識別する
2	タイトル Security goal identification 概要　セキュリティゴールを識別する
3	タイトル Threat identification 概要　脅威を識別する
4	タイトル Risk analysis 概要　リスク分析
5	タイトル Countermeasure determination 概要　対応策の決定

関連ソース

関連ステップ	ソース名	ソース種類
1.Asset identification	Security Needs Identification for Enterprise Assets	パターン
2.Security goal identification	Security Needs Identification for Enterprise Assets	パターン
3.Threat identification	STRIDE	ガイドライン
3.Threat identification	WASC	ガイドライン

図 4　方法論と関連する情報の一覧表示

図 5 は登録されているソフトウェアセキュリティ開発演習成果物の概要を表示した画面の一例である。

セキュリティ成果物名	オンラインショッピング_ミスユースケース図
成果物バージョン	0.1
成果物種類	ユースケース図
成果物登録日	2011-08-02
方法論	Methodology by Sindre and Opdahl
情報ソース	Security Needs Identification for Enterprise Assets
設計根拠	"新規利用者を登録する"では，"登録者情報"の登録を行なっている．この"登録者情報"は，「顧客とビジネスパートナーのデータ」に該当する．"登録者情報"は，ユースケース"ログインをする"，"登録者情報を確認する"，"登録者情報を編集する"，"注文を確定する"で利用されている．"注文を確定する"では，"登録者情報"の他に「法律上のデータ」として"注文情報"がある．"商品を登録する"，"商品を管理する"では，「公的なデータ」として"商品情報"がある．これら3つをアセットのゴールはパターンに従って設定する．しかし"注文を確定する"では"登録者情報"を利用できる状態にしておく必要があるため，可用性をさらに付与する．
登録者	清水啓人

図 5　ソフトウェアセキュリティ開発演習成果物に関する情報表示画面

4. 利用シナリオ

本節では学習環境の利用シナリオとして、参照情報の構築とレビューの場面を示す。

4.1. 参照情報とソフトウェアセキュリティ開発演習成果物との関連づけ

筆者らは Sindre と Opdahl により提案されたミスユースケース図を用いたセキュリティ要求獲得手法[12]に対して、Schumacher らによりまとめられたセキュリティパターン[9]、STRIDE[16]並びに WACS[17]を関連付けることにより、Sindre と Opdahl の手法を補完する手法を提案した[10]。これは参照情報において、方法論とソース情報を関連づけることにより構築される。表 1 に方法論の各ステップとそれに対応付けたパターンなどのソース情報との関連を示す。

表 1　Sindre と Opdahl によるセキュリティ要求獲得手法の各ステップと関連する情報との対応関係

Sindre と Opdahl が提案したセキュリティ要求獲得手法の各ステップ	関連付けられたソース情報
1. Identification of critical assets in the system	パターン "Security Needs Identification for Enterprise Assets"
2. Definition of the security goals for each asset	パターン "Security Needs Identification for Enterprise Assets"
3. Identification of the threats to each security goal	STRIDE and WASC
4. Identification and analysis of the risks posed by the threats	パターン "Asset Valuation," "Threat Assessment," "Vulnerability Assessment," and "Risk Determination"
5. Define the security requirements for the threats	パターン "Enterprise Security Approaches" and "Enterprise Security Service"

以下の例では入力として、ソフトウェア開発演習で作成したオンラインショッピングシステムのユースケース図を用いる。

- ステップ 1 とステップ 2

ステップ 1 ではアセットの識別を求めており、ステップ 2 では各アセットに対するセキュリティゴールの設定を求めている。これを行うために、セキュリティパターン "Security Needs Identification for Enterprise Assets"を使用し、アセット、セキュリティゴールを識別し、ユースケース図にノートに追記する形で表記を行った。その結果を図 6 に示す。

- ステップ 3

ステップ 3 では各セキュリティゴールに対して脅威の識別を求めている。それに対して、STRIDE をベースに脅威を抽出した。また、攻撃を WASC から抽出した。脅威と攻撃を記述するためにミスユースケース図を拡張した表記を提案し、図 7 のように表現する。

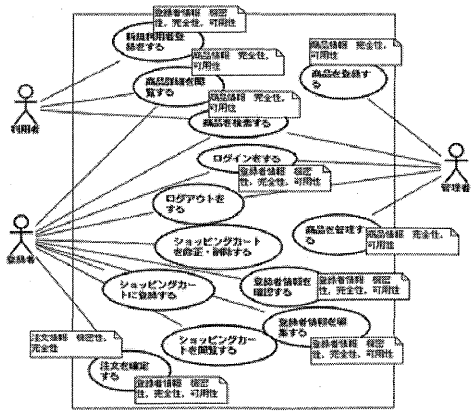

図 6　アセットとセキュリティゴールの識別

● ステップ 4

ステップ 4 では、セキュリティパターン “Asset Valuation,” “Threat Assessment,” “Vulnerability Assessment,” “Risk Determination”に基づき、前のステップで抽出したアセット、脅威、攻撃について定量的なリスク分析を行い、基準値に基づき、最終的に対応するものを抽出する。

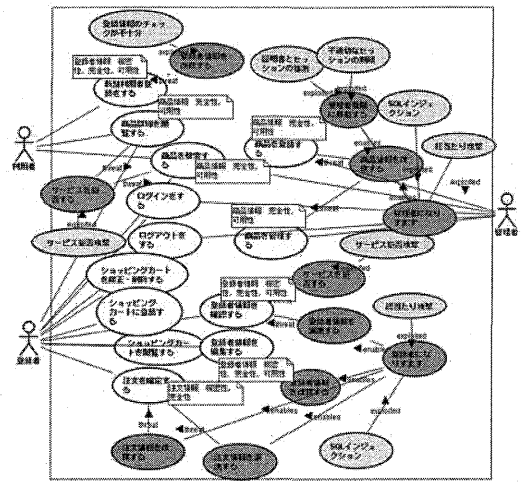

図 7　脅威と脆弱性を表現した
ミスユースケース図の拡張

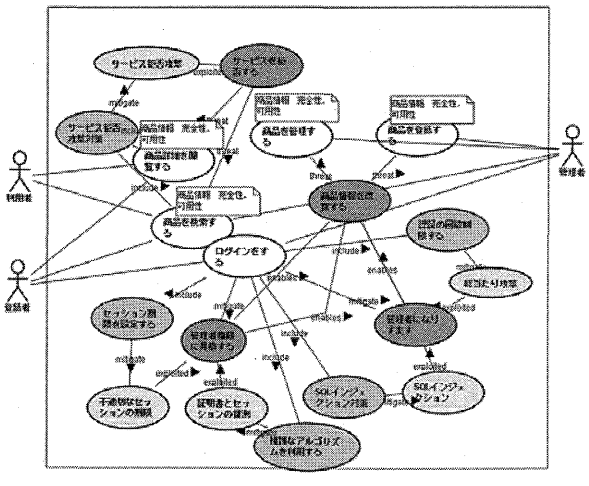

図 8　最終的なセキュリティ要求

● ステップ 5

セキュリティパターン“Enterprise Security Approaches”と“Enterprise Security Service”に従い、リスクとコストに見合う最終的なセキュリティ要求を決定する。図 8 はセキュリティ要求を表した拡張ミスユースケース図である。

4.2. ソフトウェアセキュリティ開発演習成果物に対するレビュー

本項では作成されたソフトウェアセキュリティ開発演習成果物に対するレビューのシナリオを述べる。

学習者は 4.1 項で述べた手法に従い、アセットの識別とセキュリティゴールとして必要なセキュリティの種類(機密性、完全性、可用性)を設定し、ユースケース図に反映させ根拠とともにシステムに登録することを想定する。図 5 に登録された成果物に関する情報表示画面を、図 6 に登録された成果物を示す。

図 5 に示したソフトウェアセキュリティ開発演習成果物情報の根拠には、パターンを利用してアセットの抽出やゴールを設定した理由が記述されている。この情報を他の利用者が閲覧し、レビューを行うことができる。レビューコメントの例を図 9 に示す。

概要	今回は3つのアセットを抽出しているが，管理者のログイン情報は守るべき資産ではないのか？ログイン情報はパターンの職員のデータになるのでは？
登録日	2011-08-02
登録者	橘山淳雄

図 9　ソフトウェアセキュリティ開発演習成果物に
対するレビューコメントの例

このように他の利用者がレビューコメントを記述し、学習者にフィードバックを与えることができる。学習者はコメントに基づき改版という形で成果物を登録する。図 10 にレビューコメントを反映したユースケース図を示す。

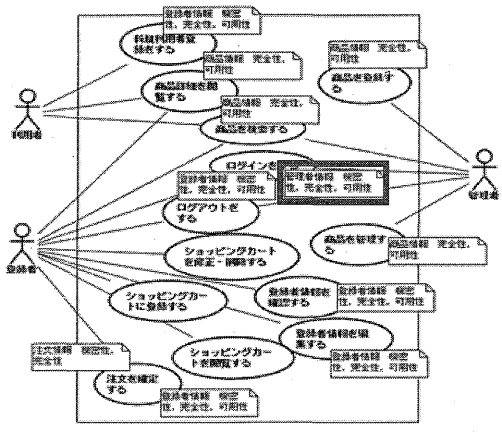

図 10　レビューコメントを反映したユースケース図

5. 関連研究

本研究に関連する研究を、ナレッジマネジメントによるセキュアソフトウェア開発支援の観点と、ソフトウェアセキュリティ学習の観点から述べる。

(1) ナレッジマネジメントアプローチによるセキュアソフトウェア開発支援に関する研究

リポジトリベースのセキュアソフトウェア開発支援環境として SHIELDS プロジェクトがある[7]。プロジェクトの目的はセキュリティの専門家の知識を表出したセキュリティモデルの蓄積と共有である。そのためのリポジトリモデルが示されている。これは本研究では参照情報に相当するものである。リポジトリモデルを見る限り、セキュリティモデル構築の背後にある根拠は扱っていない。

Barnum と McGraw はソフトウェアセキュリティの実践の基盤を形成することを目標にソフトウェアセキュリティの知識構造を提案している[1]。知識構造を "Principle," "Guideline," "Rule," "Attack pattern," "Vulnerability," "Exploit," "Historical risk"の 7 つのクラスとその関連として表現している。この研究も本研究における参照情報を詳細に規定したものと考える。

本研究では参照情報を活用して演習を行い、演習成果である成果物の蓄積、成果物と参照情報の関連づけをその根拠とともに蓄積する学習環境構築を目指すものである。

(2)ソフトウェアセキュリティ学習に関する研究

Lester はソフトウェアセキュリティの重要性が増しているという認識の下、学部レベルのソフトウェア工学コースでセキュリティを扱うことを提案している[5]。16 回の授業スケジュールの中で 2 回分セキュリティに関するトピックを扱っている(「セキュリティライフサイクル」と「設計におけるソフトウェアセキュリティ」)。しかしながら、具体的にどのような内容なのかは明らかにされていない。

6. おわりに

本稿では、ソフトウェアセキュリティ学習環境を提案した。ソフトウェア開発演習の成果物を活用するとともに、ソフトウェアセキュリティに関する方法論、パターン、標準等の知識を参照しながら、それらと関係づけて学習を進めることを可能にするものである。

今後は実装を完了するとともに、参照情報を構築し、評価を行う予定である。

謝辞

本研究の一部は科学研究費補助金 (C) 22500910 の助成のもとで行われた。ここに記して謝意を表す。

文　献

[1] Sean Barnum and Gary McGraw, Knowledge for Software Security, IEEE SECURITY&PRIVACY, Vol.3, No.2, pp.74-78, 2005.

[2] Sean Barnum and Amit Sethi, Attack Patterns as a Knowledge Resource for Building Secure Software, 2007.

[3] Atsuo Hazeyama, A Case Study of Undergraduate Group-based Software Engineering Project Course for Real World Application, Proceedings of the First International Symposium on Tangible Software Engineering Education (STANS2009), pp.39-44, 2009.

[4] 櫨山淳雄，小林祐介，成果物管理とコミュニケーション支援を連携した非同期分散ソフトウェア開発支援環境、情報処理学会ソフトウェアエンジニアリングシンポジウム 2008 ワークショップ (SES2008), pp.5-6, 2008.

[5] Cynthia Y. Lester, A practical application of software security in an undergraduate software engineering course, International Journal of Computer Science Issues, Vol.7, No.3, pp.1-10, 2010.

[6] Gary McGraw, Software Security, IEEE SECURITY&PRIVACY, Vol.2, No.2, pp.80-83, 2004.

[7] Per Hakon Meland, Shanai Ardi, Jostein Jensen, Erkuden Rios, Txus Sanchez, Nahid Shahmehri and Inger Anne Tondel, "An Architectural Foundation for Security Model Sharing and Reuse," Proc. International Conference on Availability, Reliability and Security 2009, pp.823-828, 2009.

[8] 大久保隆夫，企業におけるセキュリティ分析技術の実効性，情報処理，Vol.50, No.3, pp.230-234, 2009.

[9] Markus Schumacher, Eduardo Fernandez-Buglioni, Duane Hybertson, Frank Buschmann and Peter Sommerlad, Security Patterns: Integrating Security and Systems Engineering John Wiley & Sons, 2006.

[10] 清水啓人，櫨山淳雄，Web アプリケーション開発におけるミスユースケースを利用したセキュリティ要求獲得手法の提案，情報処理学会第 73 回全国大会，2011.

[11] Guttorm Sindre and Andreas L. Opdahl, Eliciting Security Requirements by Misuse Case, Proceedings of the 37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-Pacific 2000), IEEE CS Press, 2000.

[12] Guttorm Sindre and Andreas. L. Opdahl, Eliciting security requirements with misuse cases, Requirements Engineering Journal, Vol.10, pp.34–44, 2005.

[13] CAPEC, http://capec.mitre.org.

[14] Common Criteria, http://www.commoncriteriaportal.org/

[15] SQUARE, http://www.cert.org/sse/square/square-pubs.html.

[16] STRIDE,http://msdn.microsoft.com/ja-jp/magazine/cc163519.aspx

[17] WASC, http://www.webappsec.org/