

アスペクト指向言語を用いた UML 機能要件と CC に基づくセキュリティ要件の統合管理手法

野呂 惇[†] 加藤 洋祐[‡] 松浦 佐江子[†]

芝浦工業大学大学院 理工学研究科 電気電子情報工学専攻[†]

芝浦工業大学 システム理工学部 電子情報システム学科[‡]

1. はじめに

セキュリティは、設計・実装工程への一貫性を確保することや、要求や環境の変化に対応してその一貫性を保つ必要があるため、セキュリティ要件は対象ソフトウェア開発の要求分析段階で整理し、適切に定めなければならない。

しかし、セキュリティの関心事はそれぞれのセキュリティの特性や守るべき資産、脅威、脅威エージェント、それに対する対抗策等の様々な事を考える必要があり、その上それらは密接に関係している。さらに、想定される脅威に応じて有効な対抗策を施す必要がある。そのため、セキュリティ要求の分析にはセキュリティに関する深い知識が求められる。

我々は、情報セキュリティ国際評価基準であるコモンクリテリア (CC) をセキュリティ対策の知識とし、UML 要求分析 [1] による機能要件に対して、機能要件と分離したセキュリティ要求を分析する方法 [2] を提案してきた。

CC には対抗策となるセキュリティ機能の基礎となるコンポーネントが記述されており、脅威に対する対策としてシステムに必要な条件や操作が記載されている。また、コンポーネント同士には依存関係があり、複数のセキュリティ関心事を網羅的に分析することができる。

本稿では、機能要件と分離したセキュリティ要件をシステムの横断的関心事と捉え、アスペクト指向言語を用いてプロトタイプを生成することで 2 つの要求を統合・管理する方法を提案する。

2. 提案手法

2.1. 概要

本手法では UML 要求分析モデルの振る舞いを Java コードに、セキュリティ要件を AspectJ によりアスペクトのモジュールに変換することで、コードにアスペクトのモジュールを織り込むウィービングというアスペクト指向の仕組みを用いて機能要件とセキュリティ要件を統合する。AspectJ は、アスペクト指向の Java 実装である。

また、統合した結果をプロトタイプとして出力することで顧客・開発者の双方の分析結果の確認を可能にする。

2.2. CC を用いたセキュリティ要求定義

CC を用いた要求分析手法 [2] を用いて、アクセス制御に関するセキュリティ機能方針 (SFP) を定義する。アクセス制御に関する SFP は「システムの操作とその主体お

よび対象のセットに対して定義する規則の集合」であると CC に定義されている。したがって、機能要求分析モデルから、ユーザごとのエンティティデータに対する操作を抽出し、システムの操作とその主体および対象の集合として表形式で生成する。そして、アクセス制御に関するコンポーネント「セキュリティ属性によるアクセス制御」を基にアクセス制御したいユーザのエンティティデータの操作に対して、ユーザおよびエンティティデータのセキュリティ上の性質であるセキュリティ属性を定義し、これを用いてその操作を実行するために必要な条件を記述する。

更に、依存関係のあるコンポーネントに対するルールを検討する。例えば、「セキュリティ属性によるアクセス制御」はコンポーネント「静的属性初期化」と依存関係があるため、アクセス制御したいエンティティデータを生成する操作に対して、操作が完了したときの状態を、セキュリティ属性を初期化するルールとして定義する。ルールは「アクション前、操作実行の条件」や「アクション後、操作実行後の状態」という形式で記述する。

2.3. アスペクト指向言語によるプロトタイプ生成

まず、アクティビティ図をクラス、アクティビティ図内のノードをメソッドとして、メソッド内に次のノードに対応するメソッド呼び出しを定義することで、UML 要求分析モデルの振る舞いを Java コードに変換する。

次に、SFP 表から AspectJ のモジュールを生成する。2.2 で定義した「条件・状態の定義式」から Java コードに挿入するセキュリティ機能となる処理 (アドバイス) を、2.2 で定義したルールのタイミングである操作のアクション前およびアクション後という情報からアドバイスを挿入するタイミング (ポイントカット) を定義し、これをモジュールに変換する。

生成された Java コードとアスペクトモジュールをウィービングし、実行することで UML 要求分析モデルの振る舞いに SFP として定義したルールを挿入した Java バイトコードを生成し、生成したコードからプロトタイプ生成に必要な情報を読み取る。

[1] から出力したプロトタイプの XML に、モジュールに追加した情報をウィービングしたコードから読み取り追加し、プロトタイプを生成することで、セキュリティ要件を追加したプロトタイプを生成する。

2.4. プロトタイプによる分析結果の確認

図 1 に示す通り、生成されるプロトタイプにはシステムの処理内容、システムが表示するデータの構成を表すテーブル、ボタン・リンクを用いたユーザによる画面遷移の操作が表現される。また、システム内部のフローの条件分岐による遷移条件はボタン・リンクに表現される。

また、プロトタイプに表示されている情報は全て機能要求分析モデルまたは SFP 表に記述されている内容であり、SFP 表には機能要求分析モデル中のシステムのエン

Integrating the CC based Security Requirements and UML-based Functional Requirements using Aspect-Oriented Language

[†]Atsushi NORO [‡]Yosuke KATO [†]Saeko MATSUURA

[†]Division of Electrical Engineering and Computer Science, Graduate School of Engineering and Science, Shibaura Institute of Technology

[‡]Department of Electronic Information System, Collage of System Engineering and Science, Shibaura Institute of Technology

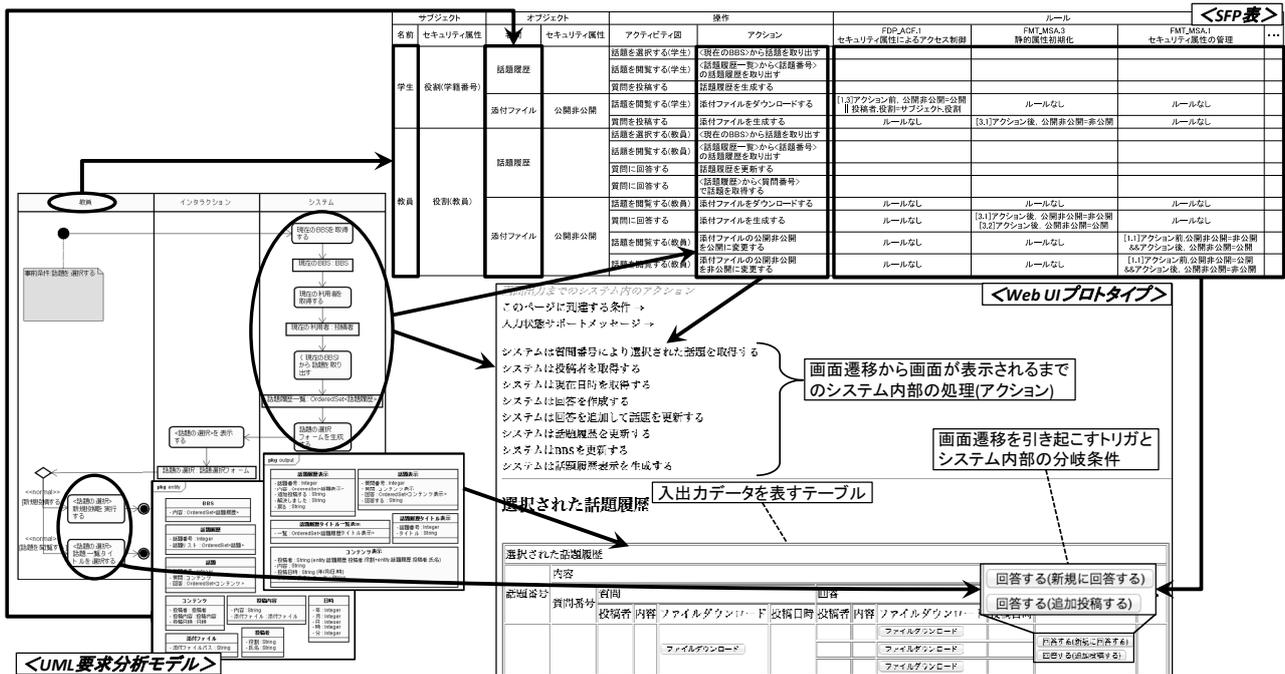


図 1 UML 要求分析モデル・SFP 表・Web UI プロトタイプの対応関係

ティティクラスに対する操作が一覧されている。したがって、プロトタイプの画面を構成している要素とモデルおよび SFP 表の要素は一意に対応づいており、開発者はプロトタイプを確認することで発見される要件の漏れや誤りまたは更なる要望に対し、その対応関係を基に分析・修正することができる。

3. 考察

本手法により統合したセキュリティ要件の管理方法について考察する。例として、大学の授業支援システムの BBS 機能に対する要求分析において、質問や回答のセット（話題）と、質問や回答に添付できるファイル（添付ファイル）のそれぞれに対して、「公開非公開」という属性を用いて学生のアクセスを制御する要求に対し、話題に対して「公開非公開」の属性によるアクセス制御を行なった際に、それに付随する「添付ファイル」も同様のアクセス制御を行なうという誤認識による、添付ファイルに対するセキュリティ要件の定義漏れに対する対処方法を考える。

図 1 に示すとおり、プロトタイプにはシステムの処理内容と画面遷移のためのボタンやリンクが表示されており、顧客は実際に画面遷移を起こすことで、「学生の添付ファイルのダウンロードがアクセス制御されていない」等の想定と異なる画面遷移を発見できる。

プロトタイプに表示されているシステムのエンティティに対する処理と SFP 表の対応関係から、開発者は、表の「学生」の「添付ファイル」に対する「ダウンロード」操作に対し、「セキュリティ属性によるアクセス制御」に基づき添付ファイルに「公開非公開」という属性を定義し、アクセス制御のためのルールを定義する。

また、依存関係のある「静的属性初期化」に対するルールとして、添付ファイルを生成している操作の後に「公開非公開」の初期値を決定するルールを定義する。

更に、「静的属性初期化」と依存関係のあるコンポーネント「セキュリティ属性の管理」に対するルールとし

て、添付ファイルのセキュリティ属性「公開非公開」の値を更新する操作のためのルールを定義する。また、「公開非公開」はセキュリティ要求分析の際に定義するので要求分析モデル中には記述されていないため、ルールに対応する操作を表中に新たに定義する必要がある。

以上より、プロトタイプで発見されたセキュリティ要件の漏れや誤りに対して、プロトタイプとモデル・SFP 表の対応関係を基にセキュリティ要件を再分析し、依存関係を用いて網羅的に分析できることが確認できた。

4. まとめ

本稿では、CC による SFP をベースにしたセキュリティ要求分析手法に対し、アスペクト指向言語を用いて機能要件とセキュリティ要件を統合し、プロトタイプを生成した。そして、分析結果に対する分析漏れや誤り、追加の要求に対して、機能要求分析モデル・SFP 表・プロトタイプの要素の対応関係により変更すべき箇所を特定し、CC の依存関係を用いてセキュリティ要件を分析することによる分離した 2 つの要件を管理する方法を示した。

本手法では、依存関係のあるコンポーネントに対してルールが必要かどうかを 1 つ 1 つ検討する必要があるため、ルールを検討する際の判断基準を記録することで、追加されたセキュリティ要求に対するルールの検討箇所を特定できるようにし、セキュリティ要件をより管理しやすくするとともに、システム稼働後の変更要求に対して本手法を適用できるよう検討する。

参考文献

[1] 小形真平, 松浦佐江子, “UML 要求分析モデルからの段階的な Web UI プロトタイプ自動生成手法”, 日本ソフトウェア科学会, コンピュータソフトウェア, Vol.27, No.2, pp.14-32 (2010).
 [2] 野呂惇, 松浦佐江子, “コモンクライテリアを用いたモデル駆動セキュリティ要求分析手法”, ソフトウェアエンジニアリングシンポジウム 2013 論文集, pp.1-6 (2013)