RO-008

# UML 要求分析モデルとコモンクライテリアに基づくセキュリティ要求分析の統合手法
# Integration Method of UML Requirement Analytical Model and Security Requirement Analysis based on Common Criteria

野呂　惇† 　小形　真平‡ 　松浦　佐江子†
Atsushi Noro 　Shinpei Ogata 　Saeko Matsuura

## 1. はじめに

システムの脆弱性はシステムに対する脅威の元となり，ビジネス上でのリスクのみならず，情報漏洩や基盤サービスの停止等の社会問題を生じさせる危険性がある．ネットワークセキュリティ技術のみならず，システムのアプリケーション特性を考慮し，ソフトウェア開発の上流工程からセキュリティ要求を定義することが必須である．

しかし，大規模かつ複雑化するシステムに対して，もれなくセキュリティ要求を分析することは，次の観点から困難である．第 1 に，一般的にシステムの開発者はセキュリティの専門家に比べてセキュリティの知識が乏しいため，どこに，どのように，どれくらいセキュリティを実現したらいいのかをうまく分析できない．第 2 に，従来の開発技術と親和性のある手法でなければ，学習コストなどの問題から新たな分析手法を導入することが難しい．第 3 にシステムの機能の要求分析とセキュリティ要求分析を同時に行うと要求分析が複雑になり，分析に漏れや誤りが生じやすく，成果物の追跡可能性が低下して手戻りが多く発生する．

本研究では，これらの問題を解決し上流工程で開発の手戻りを抑制するセキュリティ要求分析を行うための分析手法を提案する．第 1 の問題に対しては，Common Criteria (CC) [1]を分析に用いることで開発者のセキュリティの知識を補う．CC(SO/IEC 15408)とは，情報技術セキュリティの観点から，情報技術に関連した製品及びシステムが適切に設計され，その設計が正しく実装されていることを評価するための国際標準規格である．Part1 から Part3 の 3 つのパートで構成されており，Part2 にセキュリティ機能の基礎となるセキュリティ機能コンポーネント(以下コンポーネント)が記載されている．第 2 および第 3 の問題に対しては，我々が研究している Unified Modeling Language (UML)[2]を用いて要求分析を行う Web-UI プロトタイプによる機能の要求分析[3]との結合により対処する．UML とは Object Management Group (OMG)が提唱するソフトウェア開発における統一モデリング言語であり，開発現場への親和性が高い．異なるモデルを段階的に統合および変換しながら開発を進めるモデル駆動アーキテクチャ[4]に基づき，機能の要求分析とセキュリティ要求分析を分離・統合することにより，要求分析の複雑化を防ぐことができる．

さらに，統合したモデルからプロトタイプを生成し，セキュリティ要件を顧客と確認することで，セキュリティ要求の漏れを減少させることを目指す．

†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, Department of Electrical Engineering and Computer Science, Graduate School of Engineering and Science, Shibaura Institute of Technology
‡信州大学工学部情報工学科, Department of Computer Science and Engineering, Faculty of Engineering, Shinshu University

## 2. 提案手法

### 2.1 前提条件

本手法では Web-UI プロトタイプを用いたシステムの機能の要求分析を行う．Web-UI プロトタイプ手法とは，web アプリケーションを対象に業務のワークフロー分析からシステムが提供するユースケースを抽出し，UML モデルからツールによりプロトタイプ(HTML で定義された Web ページ)を自動生成しながら UML 要求仕様を定義する手法である．分析の結果，分析対象システムのアクタ，アクタが利用するユースケース，ユースケース全体の関係，ユースケースにおける振る舞いとその入出力およびエンティティのデータ，例外時の振る舞い，データの構造およびその関係と制約が UML のユースケース図，クラス図，アクティビティ図を用いて定義されていることを前提条件とする．

### 2.2 分析対象システムに対する脅威の分析

分析対象システムに対する脅威の分析は，ユースケース図を拡張し，通常のアクタに加えて誤操作等を含むシステムへの害を与えるミスユーザと，ミスユーザによるシステムの脅威であるミスユースケースを明示的にモデル化することが出来るミスユースケース図[5]を用いて行う．

まず，機能の要求分析を行ったシステムの脆弱性が悪用または誘発されるアクタをミスユーザとし，ミスユーザが利用できるユースケースのフローから，システムが守らなければならない情報である資産となるエンティティデータを抽出する．そして，抽出した資産に対してミスユーザが与える脅威を分析し，対策を講じなければならない脅威をミスユースケースとして定義する．

### 2.3 Common Criteria からの対策の選択

前項で分析された脅威に対する対策を，CC Part2 のコンポーネントから選択する．コンポーネントはその機能のカテゴリによって分類されている．分析によって脅威もある程度カテゴリ分けすることができるため，コンポーネントと照らし合わせて選択することができる．CC にはコンポーネントの依存関係が定義されているので，選択したコンポーネントと依存関係にあるコンポーネントについても脅威に対する対策として実現する．

また，コンポーネントは，システムが行わなければならない動作や条件がエレメントとして箇条書きで記されており，エレメントは割付としてコンポーネントのパラメタを特定することができる．2.1 で述べた機能の要求分析モデルとの親和性を持たせるため，エレメント，割付をそれぞれアクティビティ図のアクション，オブジェクトノードとして記述し，割付の関係をクラス図を用いて定義することで，コンポーネントをアクティビティ図とクラス図で表現することができる．

### 2.4 セキュリティ機能の定義

コンポーネントはあらゆるシステムのセキュリティを表現できるように高い抽象度で書かれている．例えば，「サブジェクト（システムの情報に対して操作を実行する主体）」や「オブジェクト（情報を格納するエンティティ）」，「操作（ユーザがシステムの情報に対して行うアクション）」のような言葉で記述されている．そのため，そのまま分析対象システムに適用することはできない．そこで，まずコンポーネントの割付を分析対象システムの要素と対応付ける．例えば，割付の「サブジェクト」は分析対象システムの分析モデル上のアクタと対応付けることができる．同じように．「オブジェクト」はエンティティクラスと，「操作」はアクティビティ図のアクションと対応付けることができる．

セキュリティ機能に必要な属性(以下セキュリティ属性)を定義することで具体化し，分析対象システムに適用可能なセキュリティ機能を定義する．

### 2.5 モデル合成

前項で定義したセキュリティ機能を Web UI プロトタイプによる要求分析モデルにマージする．具体化によって新しく定義されたセキュリティ属性は分析対象システムのクラス図上の該当するクラスに追加する．定義されたセキュリティ機能は，そのセキュリティ機能が対策となる脅威を引き起こす機能のアクティビティ図のフローにマージする．マージするポイントはキーとなるアクションやオブジェクトによって決定することができる．ただし，定義したセキュリティ機能によってはアクティビティ図のフローを変えない場合もある．この場合はセキュリティ機能が関係している部分を明示的に示すだけにとどめる．

## 3. LUMINOUS 掲示板システムへの適用例

### 3.1 LUMINOUS 掲示板システム

LUMINOUS とは本学で使用されている学習支援サイトのことである．学生は自分の履修している授業の教材のダウンロードやレポートの提出などを行うことができ，教員は教材のアップロードやレポートの作成，採点などを行うことができる．

この LUMINOUS に，学生が質問を投稿し，教員が回答を行う LUMINOUS 掲示板システム(以後掲示板システム)を追加した．掲示板システムの主な機能は学生が質問を投稿すること，教員が投稿された質問に回答すること，学生と教員が投稿された質問と回答のまとまりである話題を閲覧できることである．また，質問と回答にはファイルを添付する事ができ，閲覧時にダウンロードできる．

本稿ではこの掲示板システムに本手法を適用した例をもとに，本手法について詳しく解説していく．

### 3.2 脅威の分析

掲示板システムは既存の LUMINOUS への追加機能であるため，LUMINOUS でユーザ認証に関するセキュリティ機能は十分であると判断し，掲示板システムで想定するアクタ以外に掲示板システムを利用できるユーザは存在しないものと考える．掲示板システムのアクタは教員と学生であるが，教員は管理者であるためシステムに脅威を与えないと考え，悪意のある学生をミスユーザとして分析を行う．

ミスユーザが利用できるユースケースは「質問を投稿する」，「話題を閲覧する」であるので，これらのフローから資産となるエンティティを抽出する．「話題を閲覧する」では，アクタである学生は投稿された話題の一覧から読みたい話題を選択し，閲覧や話題に含まれる添付ファイルのダウンロードを行うことができる．ここで，話題や添付ファイルには不特定多数の学生に閲覧されてほしくない情報が記述されている可能性がある．また，話題には投稿者の情報が含まれているが，これを不特定多数の学生が閲覧できることはプライバシーの問題にかかわる．したがって，話題，添付ファイル，投稿者の情報を資産として抽出することができる．

また，これらの資産に対しミスユーザが与える脅威を分析することで図 1 のミスユースケース図を作成できる．

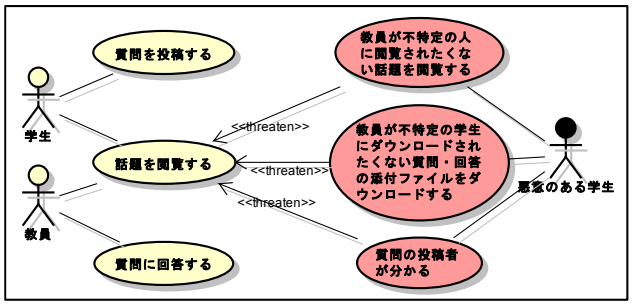


図 1　ミスユースケース図

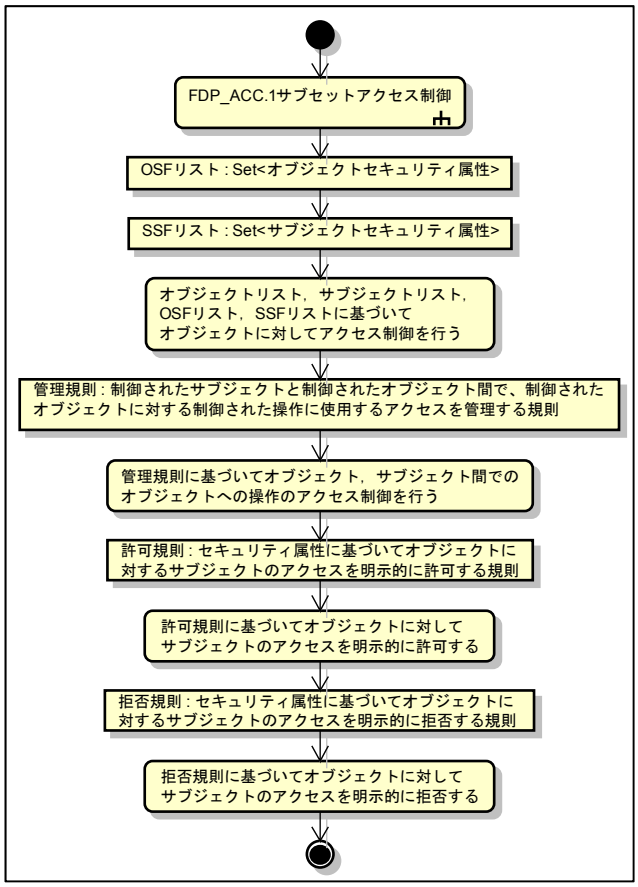


図 2　セキュリティ属性によるアクセス制御

### 3.3 対策の選択

ミスユースケースのうち，「教員が不特定の学生に閲覧されたくない話題を閲覧する」と，「教員が不特定の学生にダウンロードされたくない添付ファイルをダウンロードする」はユーザの使用するデータに関するカテゴリの脅威，「質問の投稿者が分かる」はプライバシーに関するカテゴリの脅威であると考えられる．したがって，前者は CC の「利用者データ保護クラス」，後者は CC の「プライバシークラス」から対策となるコンポーネントを選択する．

例えば，「教員が不特定の学生にダウンロードされたくない添付ファイルをダウンロードする」は，それぞれデータへのアクセスに関わる脅威であるため，「利用者データ保護クラス」の「アクセス制御機能ファミリ」の中から「セキュリティ属性によるアクセス制御」を対策となるコンポーネントとして選択する．

図 2 は「セキュリティ属性によるアクセス制御」というコンポーネントをアクティビティ図で表したものである．

また，コンポーネントは依存性という他のコンポーネントとの依存関係を持っている．図 3 はセキュリティ属性によるアクセス制御の依存関係を表したものである．コンポーネントをセキュリティ機能として実現するためには依存性のあるすべてのコンポーネントを考慮しなければならない．例えば，セキュリティ属性によるアクセス制御を脅威への対策として実現するためには，図 3 に示す 7 つのコンポーネントに記述されているセキュリティ要件を実現しなければならない．

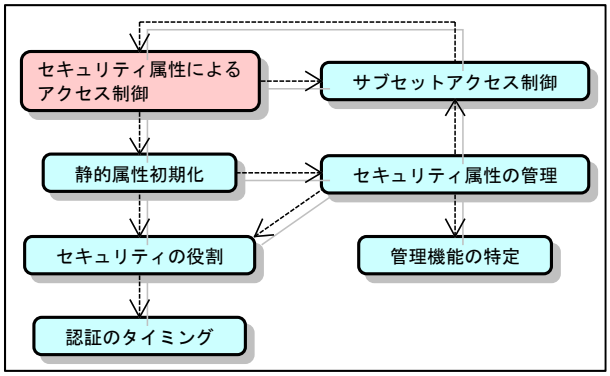


図 3　コンポーネントの依存性

### 3.4 セキュリティ機能の定義

選択したコンポーネントのアクティビティ図を掲示板システムに合わせて具体化する．表 1 は，図 2 中の「制御されたサブジェクトと制御されたオブジェクト間で，制御されたオブジェクトに対する制御された操作に使用するアクセスを管理する規則」を具体化したものである．

サブジェクトセキュリティ属性はアクタを識別している属性と対応付けることができるが，オブジェクトセキュリティ属性については，アクセス制御をかけたいオブジェクトに対して新たに作成する．

コンポーネントの抽象度を掲示板システムに合わせ，掲示板システムのセキュリティ機能として定義した結果の例を図 4 に示す．これは，添付ファイルという資産に対するアクセス制御のためのオブジェクトセキュリティ属性に初期値を与え，セキュリティ属性の値を変更するための機能である．

表 1　管理規則

| サブジェクト | サブジェクト セキュリティ属性 | オブジェクト | オブジェクト セキュリティ属性 | 操作 |
|---|---|---|---|---|
| 学生 | 役割(学籍番号) | BBS | | 質問を投稿する: 更新する<br>話題を閲覧する: 取得する |
| | | 話題履歴 | | 質問を投稿する: 生成する/更新する<br>話題を閲覧する: 参照する |
| | | 話題 | 公開 | 話題を閲覧する: 参照する |
| | | 話題 | 非公開 | 質問を投稿する: 生成する |
| | | コンテンツ | | 質問を投稿する: 生成する<br>話題を閲覧する: 参照する |
| | | 投稿内容 | 添付ファイル公開 | 話題を閲覧する: 参照する<br>話題を閲覧する: 添付ファイルをダウンロードする |
| | | 投稿内容 | 添付ファイル非公開 | 質問を投稿する: 作成する<br>話題を閲覧する: 参照する |
| | | 投稿者 | | 質問を投稿する: 取得する<br>話題を閲覧する: 取得する<br>話題を閲覧する: 参照する |
| | | 日時 | | 質問を投稿する: 取得する |
| 教員 | 役割(教員) | BBS | | 質問に回答する: 更新する<br>話題を閲覧する: 取得する |
| | | 話題履歴 | | 質問に回答する: 取得する/更新する<br>話題を閲覧する: 参照する |
| | | 話題 | 公開/非公開 | 質問に回答する: 取得する<br>話題を閲覧する: 参照する<br>話題を閲覧する: 公開/非公開を変更する |
| | | コンテンツ | | 質問に回答する: 作成する<br>話題を閲覧する: 参照する |
| | | 投稿内容 | 添付ファイル公開/非公開 | 質問に回答する: 作成する<br>話題を閲覧する: 参照する<br>話題を閲覧する: 添付ファイルをダウンロードする<br>話題を閲覧する: 添付ファイル公開/非公開を変更する<br>質問に回答する: 添付ファイル公開/非公開を変更する |
| | | 投稿者 | | 質問に回答する: 取得する<br>話題を閲覧する: 取得する<br>話題を閲覧する: 参照する |
| | | 日時 | | 質問に回答する: 取得する |

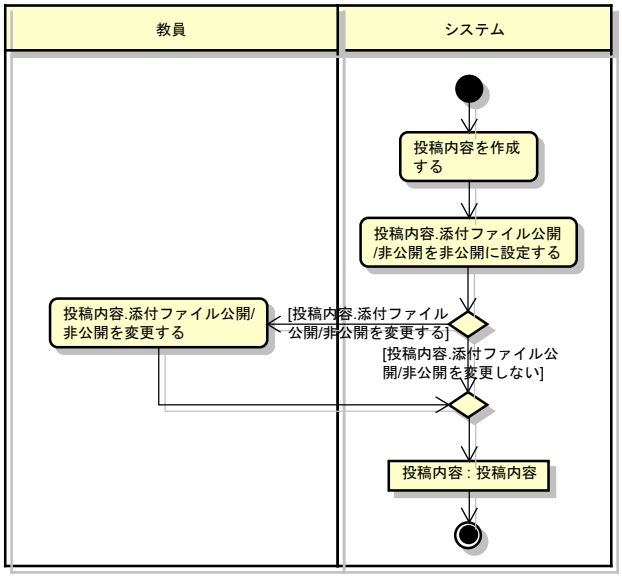


図 4　添付ファイルに関する静的属性初期化

### 3.5 モデル合成

定義したセキュリティ機能を，機能の要求分析モデルにマージする．

例えば，「添付ファイルに関する静的属性初期化」は，添付ファイルのデータをシステムが作成する際に必要なセキュリティ機能であるため，機能要求分析モデル上で添付ファイルを作成しているアクションおよびそのアクションによって生成されるオブジェクトと，「添付ファイルに関する静的属性初期化」で添付ファイルを作成しているアクションおよびそのアクションによって生成されるオブジェクトをマージのポイントとする．

合成した結果を図 5 に示す．

## 4. 考察

適用例に対し，従来手法として機能要求とセキュリティ要求を分離せず，CC などのセキュリティの知識を補うものを利用せずに掲示板システムを分析した結果である分析モデルと比較する．

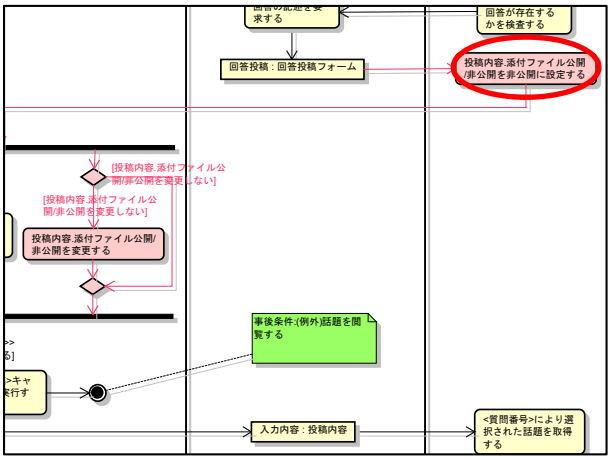


図 5　マージ後のアクティビティ図

本手法による分析では添付ファイルの静的属性初期化において，セキュリティ属性の値を設定する前にセキュリティ属性に初期値を与えている．一方で，従来手法による分析では初期値を与えるアクションは存在しなかった．これは，CC のコンポーネントにセキュリティ属性に初期値を与える記述があるため，CC を規範としている本手法では分析漏れがなかったが，従来手法では CC のような知識の拠り所がないため，分析が漏れてしまったと考えられる．

したがって，本手法により従来手法では未発見であったシステムの脆弱性を要求分析段階で発見することができ，またそれに対する対策を分析モデルに組み込むことができた．このことから，本手法により脆弱性による実装・試験段階での開発の手戻りを防ぐことができた．

## 5. 関連研究

田原らの研究[6]は，ゴール指向分析の 1 種である KAOS[7]を用いてセキュリティ要求分析を行うものである．この手法は KAOS で扱うモデルを拡張せずにセキュリティ分析を行うことができる．分析には業務知識とセキュリティの知識の両方が必要であるが，それらの知識の規範となるものは特に明示されておらず，したがって分析結果は分析者の保有する知識背景に大きく左右される．

また，網羅的な分析を行おうとするとさまざまなセキュリティの関心事をモデルに書き込むことになるため，モデルの規模が膨大になってしまい可読性を下げる結果となる．

さらに，分析結果はゴール指向分析のモデルとして導出されるので，そのまま顧客とセキュリティ要求の確認を行うことはできず，システムの機能の分析モデルにセキュリティ要求分析が反映されないため，次工程で要求の実装漏れなどが発生する懸念もある．

吉岡らの研究[8]では，ミスユースケース図を拡張し，セキュリティターゲット（以下，ST）に必要な要素をモデル上に表して整理できる．ユースケース図上でシステムとセキュリティの関係を整理し，セキュリティ要求を分析することが可能であるが，成果物がモデルであるため顧客とセキュリティ要求の確認を行うことが難しい．

本研究では CC をセキュリティ知識のベースとすることでセキュリティの知識を補い，セキュリティ要求の網羅的な分析を可能にすることで，モデルの規模が膨大になることを防ぐ．また，分析結果を機能の分析モデルに統合することで，確実に次工程にセキュリティ要求分析結果を伝えることができる．

さらに，セキュリティ要求を含むシステムの機能のプロトタイプ を生成することを目標としており，プロトタイプにより顧客とのセキュリティ要求の確認ができ，顧客の視点からセキュリティ要求の分析漏れの確認を行うことができる．

## 6. まとめと今後の方針

本手法では，CC をセキュリティ知識のベースとすることでセキュリティに関する知識を補い，セキュリティ要求分析を網羅的に行うことを可能にする．また，モデル駆動開アーキテクチャに基づき機能要求の分析モデルにセキュリティ要求分析モデルを統合することで，要求分析の複雑度を防ぎ，セキュリティ要求分析の結果を次工程に引き継げる形で表すことができる．

従来手法では未定義であった脆弱性を，本手法で発見し，対策を講じることで，実装・試験段階で起こりうる開発の手戻りを防ぐことにより，本手法の有効性を確認した．

プロトタイプの生成にはシステムのユーザとシステムのインタラクションを考えることが必要不可欠であるが，本手法ではシステムのエンティティデータに注目し，インタラクションの結果である入出力データに対して分析することによりプロトタイプの生成を行い，顧客によるセキュリティ要求の妥当性確認を行う予定である．

本研究では CC をセキュリティ知識のベースとして利用するために，CC のエレメントをフローとデータの定義から UML モデル化した．今後は，CC の体系に従いこれらのモデルを知識ベースとして管理する方法を検討する．

また，今後より多くの事例に本手法を適用し，モデル合成のパターンを検証する．そして，本手法を用いた開発者向けセキュリティ要求分析支援ツールを作成し，要求分析時でのセキュリティ要求の導入を支援する予定である．

### 参考文献

[1] 独立行政法人 情報処理推進機構,”CC/CEM バージョン 3.1 リリース 3”, http://www.ipa.go.jp/security/jisec/cc/index.html
[2] OMG,” UNIFIED MODELING LANGUAGE”, http://www.uml.org/
[3] 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).
[4] Mellor, S. J. and Scott, K. and Uhl, A. and Weise, D. ”MDA Distilled: Principles of Model-Driven Architecture”, Addison-Wesley (2004).
[5] Sindre, G and Opdahl, A. L. “Eliciting security requirements with misuse cases”, Requirements Engineering Journal, Vol.10, No.2 (2005).
[6] 田原康之, Axel van Lamsweerde, Emmanuel Letier, “KAOS によるセキュリティ要件の獲得・分析”, 情報処理 Vol.50, No.3, pp203-208 (2009).
[7] Lapouchnian, A. “Goal-oriented requirements engineering: An overview of the current research” Technical Report http://www.cs.toronto.edu/~alexei/pub/Lapouchnian-Depth.pdf, Univ. of Toronto, 2005.
[8] 吉岡信和, 田口研治, 飛田孝幸, 金子浩之, “コモンクライテリアのためのモデリング手法の提案”, 情報処理学会研究会 (2009).