GRACE TECHNICAL REPORTS

# Common Criteria に特化したセキュリティ要求分析方法論の提案

飛田孝幸, 金子浩之, 田口研治, 吉岡信和

GRACE-TR 2009–05　　2009 年 9 月 24 日

CENTER FOR GLOBAL RESEARCH IN
ADVANCED SOFTWARE SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF INFORMATICS
2-1-2 HITOTSUBASHI, CHIYODA-KU, TOKYO, JAPAN

**WWW page: http://grace-center.jp/**

# Common Criteria に特化したセキュリティ要求分析方法論の提案

飛田孝幸, 金子浩之
みずほ情報総研株式会社 情報セキュリティ評価室
{takayuki.tobita,hiroyuki.kaneko}@gene.mizuho-ir.co.jp
田口研治, 吉岡信和
国立情報学研究所 GRACE センター
{ktaguchi,nobukazu}@nii.ac.jp

2009 年 9 月 24 日

**概 要**

セキュリティ要件をシステム開発の上流工程において獲得、モデル化することは、下流において発見される可能性のある、システムの脆弱性をより少なくするための開発指標として広く認知されている。そのような方法論をシステム開発において用いることは、脆弱性の少ないシステムを開発するためにも重要であり、特にセキュリティ評価の国際標準である Common Criteria (CC) における要件定義書である Security Target (ST) を作成する際に非常に有益である。これまでにも要求工学においては KAOS や i* を用いたセキュリティ要件の獲得・モデル化方法が提案されていたが、産業界において広く利用されてはいない。その理由としては、従来の開発プロセスとそれらの方法論のミスマッチや、基礎となる意味論などが複雑なので、学習コストが非常に高いことなどが挙げられる。システム開発における標準的な設計言語である UML はシステム要件の獲得のためにユースケース図を提供している。これをセキュリティ要件獲得のために拡張したものにミスユースケース図がある。本図法はユースケース図に攻撃者や脅威、それに対する対抗策を明示的にモデル化を可能する拡張を加えたものである。本図法は、上記の 2 点の問題点をクリアしており、産業界において有効に利用することが可能である。しかし、 CC への適用を考えると、必要なモデル化要素が欠けており、必ずしも直接的に適用が可能では無い、という欠点がある。

本技術レポートでは、CC との関連において、セキュリティ要件の獲得・分析を行うためのセキュリティ要求分析方法論をミスユースケース図を拡張することで提案するものである。まず、ST のメタモデルを記述し、そのメタモデルを忠実に解釈することが可能なミスユースケース図を UML の拡張機能を用いて定義する。このことにより、両者の意味論的なミスマッチを無くすようにする。そし

て、メタモデルのどの部分をモデル化するかを段階的に獲得するプロセスを提案する。ケーススタディとして、多機能プリンタ（Multi Function Pheripheral）を用いて、本方法論におけるモデル化の方法とそのプロセスを示す。

# 1 はじめに

今日、情報システムの開発において、セキュリティを考慮するのは必須の条件である。毎日のように様々な情報システムにおいて、脆弱性が発見されており、そのための緊急パッチを当てるなどの対処法を取るのは不可欠になっている。このような状況を生む原因の一つは、情報システムの開発プロセスにおいて、セキュリティを考慮した開発手段が利用されていないことが、一つの大きな理由として挙げることが出来る。特に、セキュリティに関連する要件と、それをどのような機能として実現するかについて、要求獲得・分析フェーズにおいて十全な分析が行われない傾向があり、それがセキュリティに対する様々な対処を遅らせる原因の一つになっている。

要求工学においては、様々な方法論がセキュリティ要件の分析・獲得に関連して提案されている。しかし、実際の開発現場においては、必ずしも利用されていないのが現状である。利用されていない理由の一つとしては、提案されている方法論が実際の開発現場において利用されているものとは、大きく異なる点がある。既存の開発プロセス・開発方法論を変更するのは、非常にコストがかかるので、実際に開発現場において利用されている開発プロセスにおいて直接利用することが困難な場合、往々にして開発現場において受け入れられないのが現状である。それに対して、一般的に言うと、従来利用されているものを拡張、改良したものについては受け入れやすいという傾向がある。そのような点を考慮して、本論文においては、システム開発において広く利用されている UML (Unified Modeling Language) におけるユースケース図を拡張したセキュリティの要求分析・獲得方法論を提案する。

CC (Common Criteria) (ISO/IEC 15408) [1] はＩＴ製品及びシステムの安全性を保証するための、セキュリティ評価の国際標準である。IT 製品の製造業者にとり、 CC の取得は、製品の安全性の保証を主張するための有効な手段となっている。CC においては ST (Security Target) という評価の対象になるシステム TOE (Target Of Evaluation) のために、実装すべきセキュリティの要件に関する言明を記述した書類を作成・提出する義務がある。 CC においては、機能要件をどこまで保証しているかを表す尺度として、保証レベル EAL (Evaluation Assurance Level) と呼ばれる７段階（EAL1 から EAL7）の評価が行われている。CC において

は様々な IT 製品が認証されているが、実際のそれらの IT 製品の開発においては、必ずしもセキュリティに関する要件を要求分析工程において獲得・分析している訳ではない。多くの IT 製品においては、 CC による認証を行うために初めて、 ST としてセキュリティ要件を整理、定義しているのが製品開発の現状である。セキュアな製品開発のためには、セキュリティ要件の獲得を、他の機能要件の獲得と同時に行い、セキュリティに関する様々な要件定義を後から行う、といった製品に脆弱性をもたらすことが無い開発プロセスを導入すべきである。さらに ST の作成は、想定するセキュリティ上の課題とそれに対する対策の対応管理が困難であること、及び記述内容の他者との共有が困難であることが課題となっており、開発者が CC 取得を敬遠する一因となっている。

本技術レポートでは、我々が開発した CC をターゲットとしたセキュリティ要求分析方法論について説明を行う。本方法論は、基本的なモデリング方法論として、開発現場において広く利用されている UML (Unified Modeling Language) におけるユースケース図を拡張したものを用いる。その拡張においては、CC を意識した様々な工夫がなされている。 UML を用いる利点としては、開発現場において広く利用されているので、再教育のコストが少ない点や、その拡張可能な柔軟性、様々な図表現を用いて環境、機能要件、静的・動的なシステム要件などの記述が出来る点、視覚的に理解し易い点を上げることが出来る。ユースケース図は、要求獲得フェーズにおいて、システムの機能的な要件を分析する際に用いられており、開発対象の範囲、システムを利用するアクター、システムの提供する機能を表すユースケース、それらの間を接続する関連（association）リンクにより成り立っている。ユースケース図をセキュリティの分析に用いられるようになったのは、McDermott らによるアビューズケース図 [6] や、Sindre らによるミスユースケース図 [9] を嚆矢とする。これらの研究との比較は、第 5 章において行われる。我々の提案するミスユースケース図と ST との対応の妥当性を示すために、ST におけるモデル要素の UML プロファイルを示し、どのように示されたモデル要素が分析・獲得されるかを示す。さらに、事例に適用することで、その表現方法が効率的であるか、有効であるかを示す。

本技術レポートの構成は次のようになっている。次章において、関連技術であるユースケース（ミスユースケース）図，及び Common Criteria についてその概要を述べる。第 3 章においては、CC における セキュリティ保証の考え方について述べる。4 章において、本技術レポートにおいて提唱される CC に特化したセキュリティ要求分析方法論と本方法論を適用した事例について述べる。事例としては、多機能プリンター（Multi Function Peripheral）を用いる。第 5 章においては、他の研究との比較を

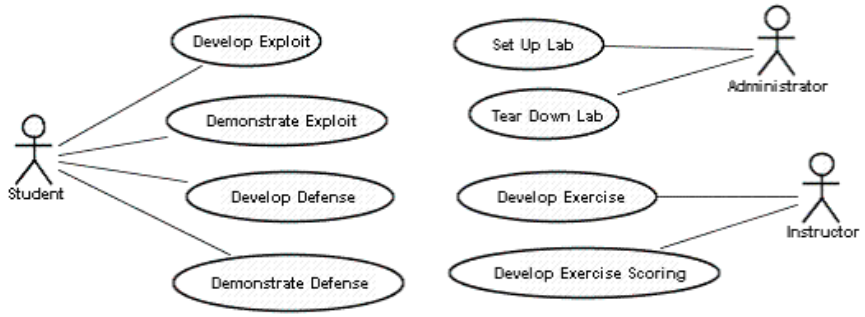


図 1: ユースケース図の例

行い、最後に第 6 章において本論文の結論について述べる。

# 2 関連技術

## 2.1 ユースケース図

ユースケース図は、システムが持ちうる機能に関してアクター (actor) と呼ばれるユーザとシステムのやりとりを図示したものである（図 1）。ユースケース図は機能要求を把握する手法として、専門用語等は用いず、顧客であるエンドユーザやその分野の専門家に分かり易く記述される。

ユースケース図（Use case Diagram）には、セキュリティに特化した様々な拡張が存在する。Sindre と Opdahl[9] によるミスユースケース図（Misuse case Diagram）においては、通常のアクターに加えて故意もしくは、意図せずにシステムに害を加えるミスユーザ（misuser）と、それによる脅威（ミスユース）を明示的にモデル化することが出来る。ここでは、ミスユースケース図のメタモデル（図 2）と、その例を示す（図 3）。

## 2.2 コモンクライテリア

今日の情報システムは、従来では達成できない業務課題をも解決する手段として大いに期待されている。その一方で、情報システムにかける投資の費用対効果は、企業経営においても、行政運営にとっても重要な関心事である。そのため、標準化されたオープンなネットワークや IT コンポーネントをできるだけ有効活用することで、許容されるコスト、期間内で情報システムを構築し、これを効果的に運用することが求められる。情報システムを複数のコンポーネントの統合により実現する際には、システム化を求める機能性以外にも、品質特性、性能特性、運用性などの非機能的

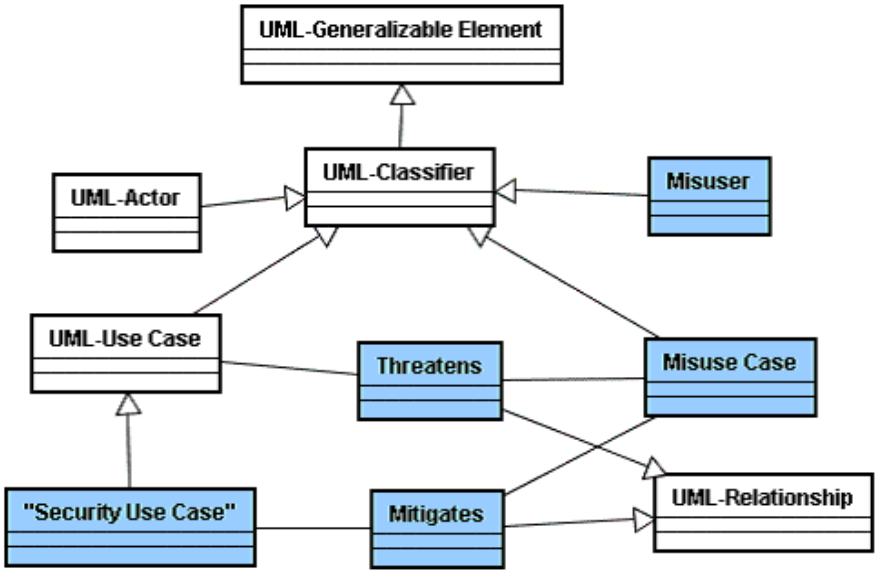


図 2: ミスユースケース図のメタモデル

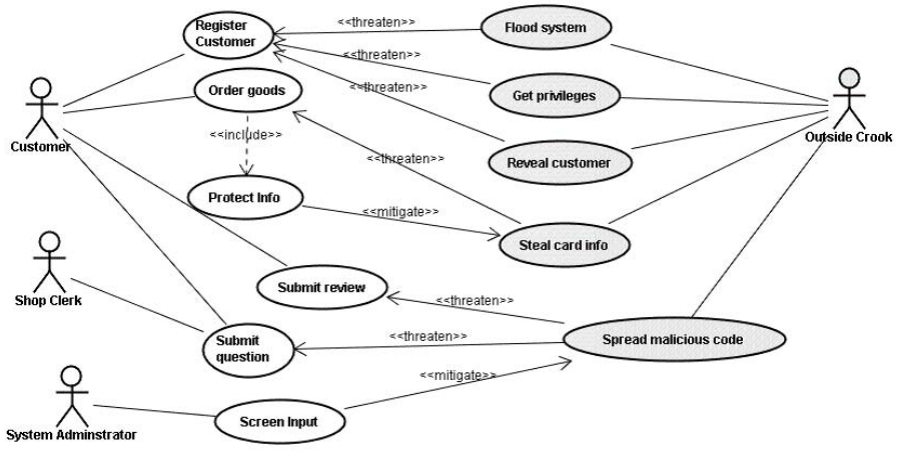


図 3: ミスユースケース図の例

な特性が求められ、その分析、設計が重要となる。セキュリティ要求は、これらの非機能要求の一部と考えられる。たとえば、システムで扱う重要な情報を資産と捉え、その特性に合わせて秘匿性、完全性、プライバシーなどを求める要求や、いつでもそのシステムサービスを提供し続ける可用性の要求など、情報システムを意図した目的で運用する際の妨げとなるリスクを低減もしくは除去するために、セキュリティ要求が規定される。これらのセキュリティ要求のうち、IT を使って実現する部分（セキュリ

ティ機能を含む）を開発する場合、その開発した部分の信頼性が保証されていることを評価するための国際標準が CC である。特に、IT に関する明確なセキュリティ要求が存在し、その要件を満たす機能を実現する主体（開発者）が明確である場合、ここにターゲットを絞ってセキュリティ要求が満たされることを、その主体が保証したい場合に、CC の適用効果が高まる。

CC の規格文書 は 3 つのパートで構成されている。

- パート 1：概説と一般モデル　評価対象（TOE:Target of Evaluation）と運用環境を正確にモデル化し、資産（Asset）、脅威（Threat）及び対抗策（Objective）によるセキュリティの概念と関係に基づいて、TOE の機能要件（Functional Requirement）と保証要件（Assurance Requirement）に関する評価の枠組みをセキュリティターゲット（ST: Security Target）として定義すること、及び ST に基づく TOE 評価の一般モデル

- パート 2：セキュリティ機能コンポーネント　セキュリティ機能要件の全体像（パラダイム）、11 のクラスからなる機能要件のカタログ情報

- パート 3：セキュリティ保証コンポーネント　セキュリティ保証のアプローチに関する全体像（パラダイム）、評価保証レベル（EAL：Evaluation Assurance Level）の定義、ST やプロテクションプロファイル（PP：Protection Profile。特定の製品種別に対するセキュリティニーズについて、実装に依存せず、ST の雛形として用いることができる構成でまとめた文書）を含む 8 クラスからなる保証要件のカタログ情報

また、CC に基づく評価方法論を示す CEM（Common methodology for Information Technology Security Evaluation） と呼ばれる規格文書がある。CEM は、CC パート 3 の各保証要件に対して、評価者が実施すべき評価アクションを具体的に示したものである。評価者は、ST で定義された評価保証レベルの保証コンポーネントに対し、CEM に定義されるより詳細な評価単位であるワークユニット毎に評価を行う。

## 2.3　コモンクライテリアに基づく保証と評価

情報システムが要求する重要なセキュリティ機能性を実装するためには、この機能性を持つ汎用製品をコンポーネントとして活用してシステムを組むことが多い。そこで、これらの製品の開発者に対し、所定の要

求を満たすセキュリティ機能性やセキュリティ保証に関する信頼性の確保を求めるために、各国で IT セキュリティに関する第三者による評価・認証制度の整備が進んでいる。日本では「IT セキュリティ評価及び認証制度」（JISEC：Japan Information Technology Security Evaluation and Certification Scheme）が運用される。また、各国の評価・認証制度で取得した CC 認証を認め合う CC 相互承認協定（CCRA：CC Recognition Arrangement）により、自国認証以外の幅広い CC 認証済み製品の国際相互流通を可能としている。CC 認証済み製品としては、OS、DBMS、ファイアウォール、ネットワークコンポーネント、データ保護、アクセス制御、認証などのメカニズムを提供するコンポーネント、スマートカード、セキュリティを重視する LSI など、多くの分野の 1000 を超える製品が各国認証制度の Web サイト等でリストされている．これらの分野の主要製品の多くは、既に CC 評価・認証を取得している。なお、評価者は、セキュリティ保証の実務に関する豊富な経験と評価ノウハウを有し、かつ制度から認められた組織（認定された評価機関）に属し、中立公平な第三者の立場から、設計評価、試験評価、侵入検査などを含むセキュリティの評価を実施する。

ここでは、CC に基づいた評価を受ける製品開発者から見た典型的な CC の適用例を示す。CC や CEM の規格文書は、評価を行う側の視点で記述されている。また、パート 2 セキュリティ機能要件やパート 3 セキュリティ保証要件の規定は、開発者の通常の理解よりも抽象的な表現で示されている。そこで、製品開発者は、CC や CEM の規定を開発者側への要求事項として読み替え、対象製品への CC 適用を行う場合の具体的な実施事項を検討し、CC に基づく保証の対応方針を明らかにする。その上で、以下に示すアプローチで、製品セキュリティの特性と構造の記述を、自然言語で ST として作成することから開始する。

(1) 利用者視点で製品に求められるセキュリティ要求を収集整理する。適用すべき PP があれば、その内容を分析する。そのほか、製品戦略上、重要と位置付けたセキュリティ要求についても検討する。これらを踏まえ、製品セキュリティのコンセプトを定義する。

(2) 評価の効果と保証の実施可能性を踏まえ、評価対象の範囲と EAL を定める。

(3) 保護対象とする資産に関する脅威モデルを定義・分析し、想定する特定の運用環境における対抗策を定義する。定義した脅威と対抗策の関係は、この製品を採用する利用者が納得できる論理構造となっていることを確認する。

(4) 評価対象にて IT メカニズムとして対抗策を実現するための機能要件を、パート 2 からの機能要件の選択、拡張により定義する。

(5) 機能要件を評価対象においてどのような機能として実現するかの要約を仕様化する

(6) これらの検討結果を踏まえ、図 4 にある ST 文書の構造通りに記述する。

ST を作成したのち、EAL の定義にしたがった個々の保証コンポーネント（開発、ガイダンス文書、ライフサイクルサポート、テストの各クラスのうち、設定された EAL で満たすべき保証コンポーネント。図 5 に CC パート 3 で定義する保証コンポーネントの構成を示す）が要求する評価エビデンスや、評価者がセキュリティ機能のテストや侵入検査を実施するためのテスト環境を準備する。脆弱性評定クラスに対しては、開発者が準備するエビデンスはテスト環境以外に存在しない。ただし、EAL のすべての保証コンポーネントを満たすことを主張するためには、評価を受ける前に、顕在化する脆弱性が存在しないことを確認しておく必要がある。具体的には、最初に評価対象の動作に影響を与える公知の脆弱性情報を収集する。次に、評価対象の仕様や設計・開発の過程で脆弱性が組み込まれる可能性がある部分を洗い出し、これらの脆弱性が評価対象において対策済みであることを確認する。また、必要に応じて脆弱性の残存有無を確認するために、開発者自身による侵入検査を実施することが望ましい。

このように、ST をはじめ、評価エビデンスの最終内容は、ライフサイクルサポートの一部のエビデンスを除き、開発が完了し出荷対象となる TOE を対象とした、完成されたものである必要がある。CC の評価においては、セキュリティの要求獲得・記述・分析フェーズのエビデンスは求められない。一方、製品の導入を検討する利用者は、その利用者のニーズに合った製品構成であり、意図する使い方が可能かどうかの確認のみでなく、ある特定の脅威への対抗策を具備しているかなど、個別のセキュリティ要求についても確認するかもしれない。また、利用者の想定する運用環境において、その製品がこれらのセキュリティ要求を満たすことが検証済みであることを期待するかもしれない。これらの確認は、ST を拠り所として行われることから、ST を作成する際には、利用者の検討候補となり得る評価対象の構成や、想定する運用環境に合致した検証環境を特定することが不可欠である。CC 評価では、ST に定義された内容が、その製品の導入を検討する利用者からみて、評価範囲や評価したセキュリティ機能について誤解を生じることがない記述かどうかの観点からも検査する。

## 3　セキュリティ保証における課題

CC に基づく評価では、最初に ST を検査し、ターゲットとする TOE の範囲の確認、及び主張するセキュリティ構造の妥当性（開発者側が想定

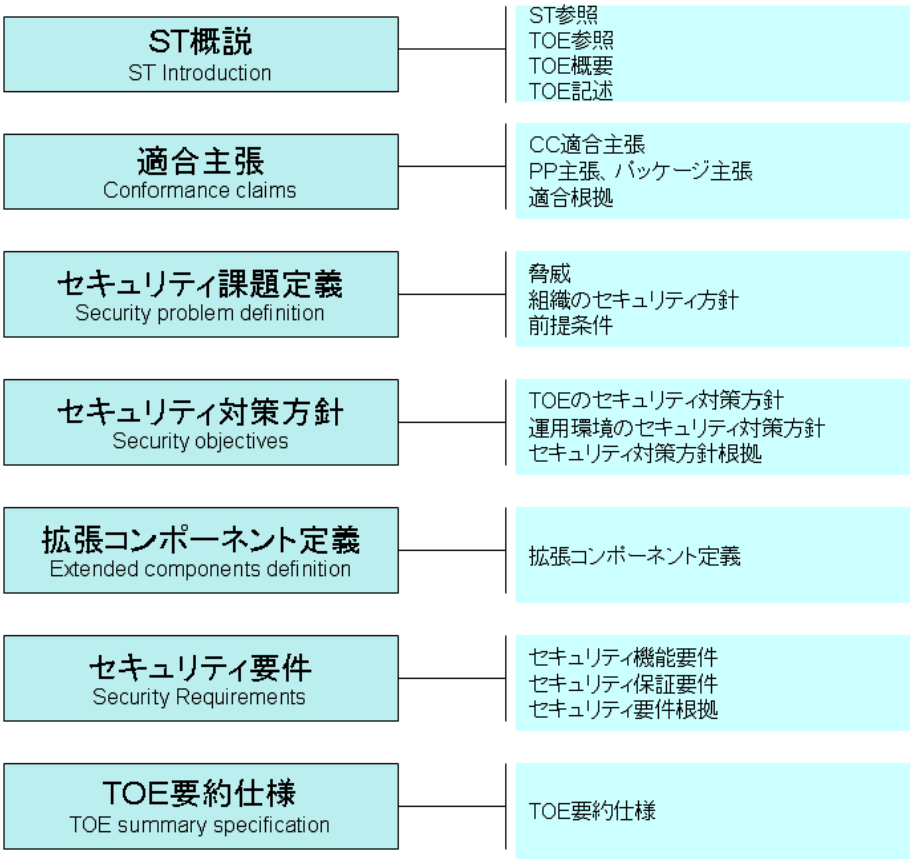


図 4: ST の構造

した妥当な脅威、この脅威に対抗するための対策、対策のために必要となるセキュリティ機能要件、機能要件を実現するための仕様の要約、について各々の相互関係の完全性及び正当性）を評価する。その後、EAL で指定された範囲の開発（ADV）クラスを始めとするその他保証コンポーネントの評価を行う。これまでの CC の評価実務の経験から、開発者の多くは、セキュリティ要件定義に至るセキュリティ構造分析を、要求獲得・記述・分析のフェーズで実施する例は少ない。多くの場合、仕様設計の段階もしくは実装設計の段階において ST を作成し、必要な設計フィードバックをかける方針で進める例が多い。このこと自体が、CC の要求を満たしていないということではないが、セキュリティの要求分析や要求管理が適正なタイミング、方法で行われないことで、本来、上流工程で対応されるべき問題（要求との不一致や仕様バグ等）が残留するおそれがある。以下では、製品の開発現場において、セキュリティ要求の規定に関連した CC 対応の現状の課題とその要因を示す。

| 保証クラス | 保証ファミリ | 省略名 | 評価保証レベル(EAL)/保証コンポーネント | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | EAL1 | EAL2 | EAL3 | EAL4 | EAL5 | EAL6 | EAL7 |
| ASEクラス | ST概説 | ASE_INT | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| セキュリティターゲット評価 | 適合主張 | ASE_CCL | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | セキュリティ課題定義 | ASE_SPD | | 1 | 1 | 1 | 1 | 1 | 1 |
| | セキュリティ対策方針 | ASE_OBJ | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| | 拡張コンポーネント定義 | ASE_ECD | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | セキュリティ要件 | ASE_REQ | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| | TOE要約仕様 | ASE_TSS | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ADVクラス | セキュリティアーキテクチャ | ADV_ARC | | 1 | 1 | 1 | 1 | 1 | 1 |
| 開発 | 機能仕様 | ADV_FSP | 1 | 2 | 3 | 4 | 5 | 5 | 6 |
| | 実装表現 | ADV_IMP | | | | 1 | 1 | 2 | 2 |
| | TSF内部構造 | ADV_INT | | | | | 2 | 3 | 3 |
| | セキュリティ方針モデル化 | ADV_SPM | | | | | | 1 | 1 |
| | TOE設計 | ADV_TDS | | 1 | 2 | 3 | 4 | 5 | 6 |
| AGDクラス | 利用者操作ガイダンス | AGD_OPE | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ガイダンス | 準備手続 | AGD_PRE | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ALCクラス | CM能力 | ALC_CMC | 1 | 2 | 3 | 4 | 4 | 5 | 5 |
| ライフサイクルサポート | CM範囲 | ALC_CMS | 1 | 2 | 3 | 4 | 5 | 5 | 5 |
| | 配付 | ALC_DEL | | 1 | 1 | 1 | 1 | 1 | 1 |
| | 開発セキュリティ | ALC_DVS | | | 1 | 1 | 1 | 2 | 2 |
| | 欠陥修正 | ALC_FLR | | | | | | | |
| | ライフサイクル定義 | ALC_LCD | | | 1 | 1 | 1 | 1 | 2 |
| | ツールと技法 | ALC_TAT | | | | 1 | 2 | 3 | 3 |
| ATEクラス | カバレージ | ATE_COV | | 1 | 2 | 2 | 2 | 3 | 3 |
| テスト | 深さ | ATE_DPT | | | 1 | 2 | 3 | 3 | 4 |
| | 機能テスト | ATE_FUN | | 1 | 1 | 1 | 1 | 2 | 2 |
| | 独立テスト | ATE_IND | 1 | 2 | 2 | 2 | 2 | 2 | 3 |
| AVAクラス　脆弱性評定 | 脆弱性分析 | AVA_VAN | 1 | 2 | 2 | 3 | 4 | 5 | 5 |

図 5: EAL と保証コンポーネントの相互参照（CC パート 3 より）

### 3.1　CC 導入段階

CC の導入段階では、開発者は初めて ST を作成することが多く、ST の定義内容の内部一貫性の欠如や、開発エビデンスとの対応の不整合が起こりやすい。CC は図 4 の ST の構造と記述内容を規定しており、TOE で扱う脅威、対策、セキュリティ要件について、論理矛盾のない識別単位で構成し、それぞれの構成要素の簡潔明快な説明（CC ではステートメントと呼ぶ）、及び完全性、正当性に関する根拠を自然言語で記述することを求めている。ただし、CC は ST を作成するための元となる要求分析や脅威分析の手法を定めていない。また、評価すべき対象とするセキュリティ要件について、PP や市場要求を踏まえ、要件の取捨選択等の現実的な調整が必要となる。しかしながら、実際には、開発者は、製品の仕様要求、セキュリティ要求、及び CC で評価するセキュリティ要求間の対応関係や論理構造を分析・管理・共有する手段を持たないケースが多い。これが ST 定義の不整合や、ST 作者以外の設計者が担当する開発エビデンスの対応不整合の原因となる。

### 3.2　要求工学的アプローチのセキュリティ保証への適用の方向性

このような CC の保証とセキュリティ要求分析に関する現状の課題、及び CC に基づいたセキュリティ保証活動の実効性を高めるためにセキュリ

ティ要求分析がどのように適用されるべきかについての方向性を以下に示す。

### 3.2.1 CC に最適化した要求のモデリング方法論とすること

CC では ST に基づいて保証を主張することが原則であることから、CC のセキュリティ構造とモデルの構造との対応関係、及び表現における親和性は必須要件である。TOE が満たすべきセキュリティ要件は、保護すべき資産と脅威の関係をベースとしたモデル化が必要である。ただし、ST は設計書ではなく評価のための定義書である。CC に基づいて評価できることが重要であり、セキュリティ要求仕様の全体と完全に一致する必要はない。一方、利用者が自身の要求を満たすことが確認できるレベルの正確なステートメント表現が求められ、ST 上の識別ラベルを使って表現できると分析の効率性が高まる。また、モデリング方法論は、開発エビデンスとの対応関係を維持するために、製品仕様の分析モデルと相互参照可能であると開発エビデンスとの対応が取りやすく CC の適用のトータルコストが削減できる。

### 3.2.2 製品開発現場での導入に負担が少ないこと

要求分析の方法論は、過度なコストをかけることなく製品開発現場に導入できる必要がある。ここでのコストとは、方法論の理解・習得にかかる教育コスト、方法論を実現するサポートツールの操作性、コミュニケーションやネゴシエーションに要する総時間、製品設計・開発に対するインパクト（効果または負担）に伴うコスト、知識移転に要するコストなどを想定する。

### 3.2.3 要求管理が製品ライフサイクル全体で維持可能であること

製造者の責務範囲で行う実際のセキュリティ保証は、製品ライフサイクル全体が対象となる。初期の製品化範囲のみでなく、脅威事象の変化に対応する TOE の修正、機能拡張などの製品仕様の変更などに伴うセキュリティ要求の更新情報を維持し、対応すべき保証の影響箇所を正確に把握するために、要求モデル記述の修正と分析を繰り返し実行でき、開発フェーズ間のモデルのトレーサビリティを保つ必要がある。そのため、製品の企画・開発・保守・廃棄のプロセス全般において、モデルの一貫性が維持できるように管理する必要がある。

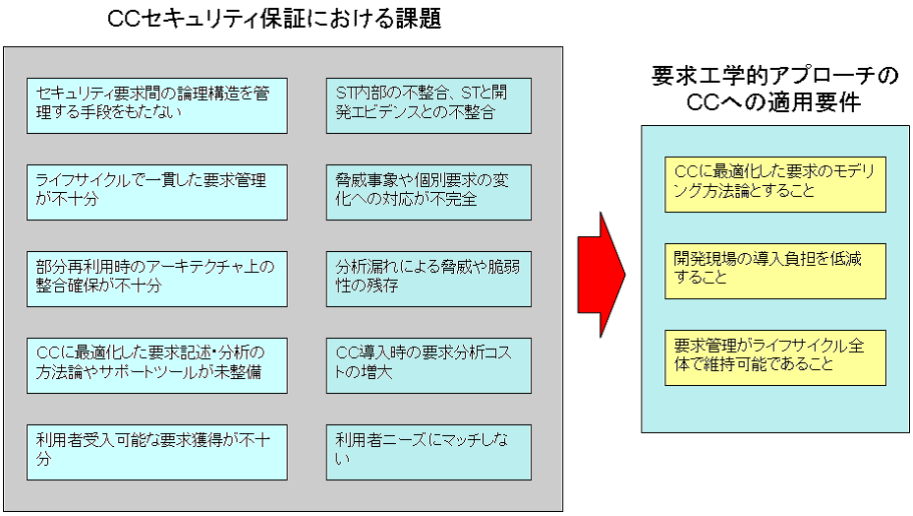


図 6: CC によるセキュリティ保証の課題と要求工学的アプローチの適用要件

# 4　セキュリティ要求分析方法論

CC との親和性を意識したモデリング手法として、UML（Unified Modeling Language）を拡張する場合の例を示す。UML を用いる利点としては、開発現場において既に広く利用されているため、導入コストがかからないこと、拡張に対して柔軟性を持ち、かつ視覚的にも理解しやすく、分析におけるコミュニケーションが容易であることが挙げられる。このモデリング手法により製品のもつセキュリティ機能（function）と資産（asset）の観点からそれに対する脅威と対策を分析し、明示的にモデル化することが可能となる。

本章においては、本技術レポートにおいて提唱するセキュリティ要求分析方法論について述べる。本方法論の特徴の一つは、通常の IT 製品のシステム開発における要求分析・獲得段階におけるセキュリティ要件の分析・獲得を CC の取得に関して ST を作成する際の実際の開発プロセスにおいて用いられる（セキュリティ要件を含む）要件定義書と CC を取得するための ST では、定義書の形式が異なるだけではなく、その対象、記述方法なども異なる。例えば、 ST において対象となる TOE は必ずしも開発されたシステム全体ではなく、その一部である場合もある。また、セキュリティ機能の網羅する範囲も異なってくる。我々の提案する方法論は、開発プロセスにおいて ST を作成するのと同様な概念を用いて行うので、開発プロセスとセキュリティ保証両方を支援することを目的としている。

## 4.1　従来のユースケース図との親和性

従来のユースケース図の構成要素は、利用者や管理者等のアクター（actor）と関連するユースケースであるが、特定のシステムについてのユースケース図には、以下が潜在的に存在するはずである。

- ユースケースで取り扱われるデータ（data）
- ユースケースを提供する機能（function）

ここでは、それらの要素を用いて、特定のシステム（または製品）のセキュリティ機能の妥当性について分析することを考える。システムまたは製品のユースケース図では、ユースケースを提供する function がセキュリティ機能そのもの、もしくはセキュリティ機能を含む場合、「このような操作は望ましくない」といった観点でユースケースが示されることはない。一方、セキュリティ機能の妥当性を分析する場合、望ましくない操作等（いわゆる脅威）に対抗するための防御策としての効果と、ビジネス上の目的を実現する機能としての合理性、利便性、実現可能性（コスト効果を含む）とのバランスが重要である。そのため、誰が引き起こすどのような脅威を想定するのか、また各セキュリティ機能はどの脅威に対して対抗すると認められるのかの関係を、モデル上で分析することを考える。またここでは、分析の成果として、2.2 節で説明した CC 認証制度の取得に必要となるセキュリティターゲット（ST）作成を目標として説明する。ST 作成を目標とすることにより、脅威（threat）、TOE のセキュリティ対策方針（objective）、及び評価のターゲットとすることが明示されたセキュリティ要件の関係から成るセキュリティの構造を明らかにすることができる。

## 4.2　フェーズ１：前提となるセキュリティに関する問題意識

ここでは、ユースケースに対して data の中で保護すべき資産を保護資産（asset）とする。また、asset とした際に想定する threat を定義し、threat から asset を保護する為に有効と想定されるセキュリティ機能（security function）を定義する。この際、マーケティングの結果、特定のシステム（または製品）に対して公知の攻撃手段等、市場が求めている function を導く「考慮すべき threat」として定義する部分、企画において市場への訴求効果のある技術を「実装すべき security function」として定義する部分など、企画段階・設計上流段階におけるコンセプト設計を反映したモデルができる。メタモデルにすると図 7 フェーズ 1 のメタモデル図となる。

3 つの要素 security function、asset、threat の洗い出しはシステム（及び製品）の種類や開発方針等により異なるが、どの要素も潜在的に他の 2 つ

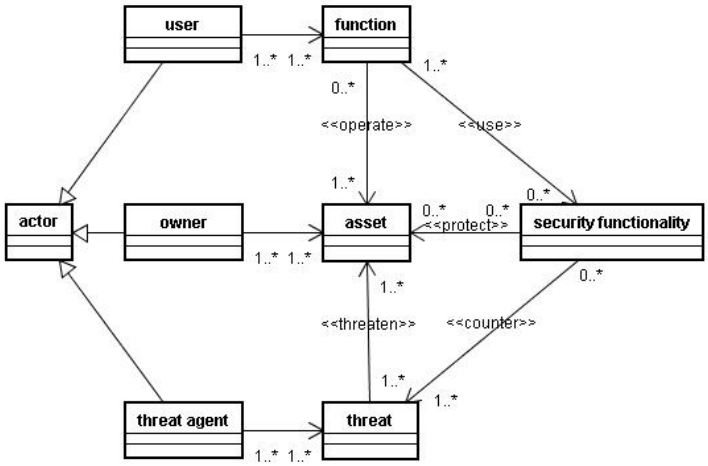


図 7: フェーズ 1 のメタモデル図

の要素に関連している。例えばセキュリティに関する特定の機能（security function）は、特定の悪意のある行為（threat）から保護すべき何らかの資産（asset）を保護する為に作られているはずである。どの要素をきっかけに記載してもよいが、必ず関連する他の 2 要素が想定できるはずである。このフェーズでのリンクは完全である必要は無いが、CC 認証制度を考慮した場合、要素が孤立するようであれば、それは ST に記載する要素では無く、図からも削除するべきである。このフェーズは ST に記載しようとしている機能の意味を整理するフェーズであると言える。なお本フェーズの分析により、CC 認証制度の ST の 1 章へ記述に対応した材料が揃うことになる。

表 1: 基本用語の説明

| | |
|---|---|
| 機能（function） | TOE に含めようと考えている機能 |
| 資産（asset） | 機能により保護される対象データ |
| 脅威（threat） | 想定していた資産への不正 |
| セキュリティ機能 (security function) | function の中で asset を保護する為の機能、security function により利用される asset も記載する。 |

### 4.3 フェーズ２：課題定義と対策

フェーズ２では、フェーズ１で洗い出した security function、asset、threat を用いて図 8 のメタモデルに従い分析を行う。このフェーズでは簡単の為に function は図示しなくても良い。ここでは基本的に以下のステップを繰り返し、分析する。なお以下のステップはどこから始めてもよく特に順番は無い。

表 2: フェーズ２のプロセス

| | |
|---|---|
| ステップ１ | threat に対する objective を記述し、その objective を満たしている security function、及び対抗している threat に接続する |
| ステップ２ | objective により新たな security function や asset が必要となった場合記述する。 |
| ステップ３ | asset に対し新たなら threat を想定した場合、記述し asset に接続する。 |

このサイクルの中で、threat に対して TOE で実装している (実装を予定している) 機能での対策が困難である場合、環境のセキュリティ対策方針 (operational environment) を記述する。また、企業のセキュリティ方針 (policy) を想定する場合は、policy を threat と同等に扱い（threat と異なり対策することにより新たな policy が想定される事は無いが）objective、もしくは operational environment により対策する。

上記のサイクルを繰り返し新たな要素が出てこなくなった段階で本フェーズは終了となる。この段階で、operational environment によってのみ対策している threat は、前提条件 (assumption) として記述する。本フェーズが終了した段階で ST の全体像を掴む為に必要であると言われる 4 章までが記述可能である。

### 4.4 フェーズ３：セキュリティ機能要件

フェーズ３では図 8 に示したメタモデル図に対して、objective を満たし、かつ対応する security function のメカニズムに関する要求を構成するセキュリティ機能要件（SFR）を抽出する。抽出した SFR は、図 9 のように、対応する security function と objective との間に、それぞれリレーションで関連付ける。これらの SFR を抽出する作業は、ST の 6 章以降を記述する為に必要であり、CC パート 2 にカタログされている SFR を参考として抽出する。SFR を抽出する際には、まず objective を満たす主となる機能要件を CC パート 2 から抽出する。たとえば、主体の識別

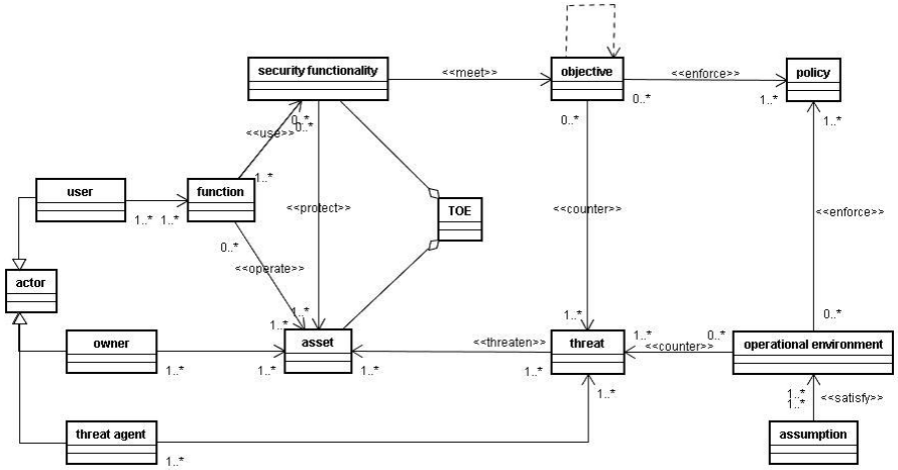


図 8: フェーズ 2 のメタモデル図

認証を求める objective であれば、CC パート 2 の FIA クラスの中から識別（FIA_UID ファミリ）及び認証（FIA_UAU ファミリ）の中から、この objective を満足する主となる機能コンポーネントとなる SFR を選ぶ。CC パート 2 にカタログされた各 SFR は、その SFR とセットでメカニズムを構成する傾向がある他の SFR が示されている（CC では「依存性」という）。また、依存性の規定はなくても、この SFR で扱う属性やデータを管理するために通常セットで定義する SFR なども参照できる。これらの参考情報や、他製品の ST に示された SFR 群のサンプルなどを参考として、objective を満たすために構成すべき SFR のセットを抽出していく。SFR 抽出の段階で objective を満たすために十分な SFR が無い場合は、通常の ST 作成と同様に新たに SFR を定義し 5 章に記述する必要がある。SFR のセットで objective が十分満たされること、過度な要件となっていないことについては、SFR セットで期待される効果と制約（及び制約により新たに引き起こされる脅威や脆弱性がないこと）を検討し、コスト的にも妥当であることを確認する。SFR 抽出の結果、その SFR を実現する security function について、当初想定した security function の見直しが必要となることもある。その場合は、図 9 のメタモデルの security function を修正し、関連する asset への影響や、新たな threat への影響をモデル上で確認することが必要であり、釣り合いの取れたモデルとなるまでブラッシュアップを図っていく。

フェーズ 1 〜フェーズ 3 が完了すると objective と SFR の関連を ST の 6 章に、security function と SFR の関連を ST の 7 章にそのまま適用することができ、ST の記述に必要な材料が整ったと言える。

表 3: フェーズ 2 の用語

| | |
|---|---|
| TOE のセキュリティ対策方針（objective） | threat や policy に対する対抗策の方針 security function により実現可能な範囲で記述 |
| 環境のセキュリティ対策方針 (operational environment) | threat や policy、assumption に対する対抗策の方針 TOE の機能ではなく、運用等で対抗する方針を記述 |
| 組織のセキュリティ対策方針（policy） | 組織の規定等で、運用やセキュリティ強度が定義されているものを記述 |
| 前提条件 (assumption) | 運用や環境の前提条件により threat に対応するものを記述。 operational environment による実現が必要 |

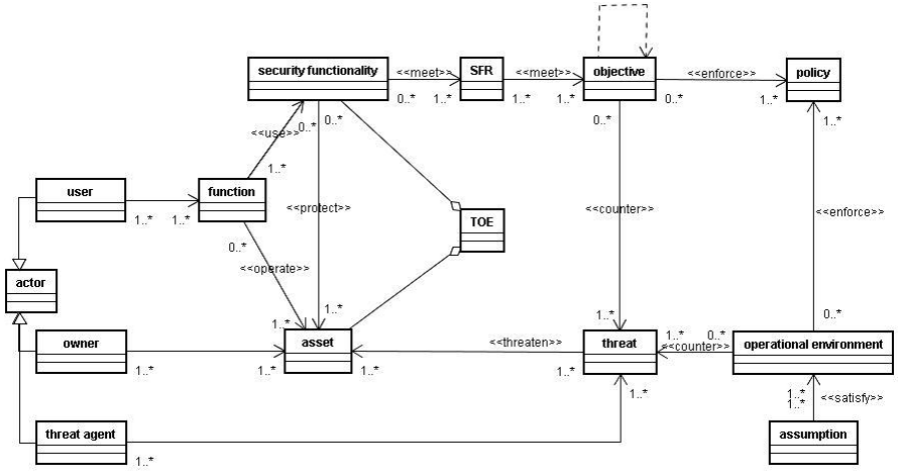


図 9: フェーズ 3 のメタモデル図

## 4.5 具体的な例

CC 認証実績の多い複合型プリンタ（MFP：Multi Function Peripheral）[4] のセキュリティ文書印刷機能（Security Print Function）という機能に対して分析を行う。なおここでは簡単の為に一つの機能のみ示すが、文献 [4] には他の機能も説明されている。セキュリティ文書印刷機能とは、プリントデータと共にセキュリティ文書パスワードを受信した場合、画像ファイルを印刷待機状態で保管し、パネルからの印刷指示とパスワード入力により印刷を実行する機能であり、これよりクライアント PC からのプリント行為において、機密性の高いプリントデータが、印刷された状態で

表 4: フェーズ 3 の用語

| | |
|---|---|
| セキュリティ機能要件 (SFR) | security function により実現され、objective を実現するために十分である機能要件 |

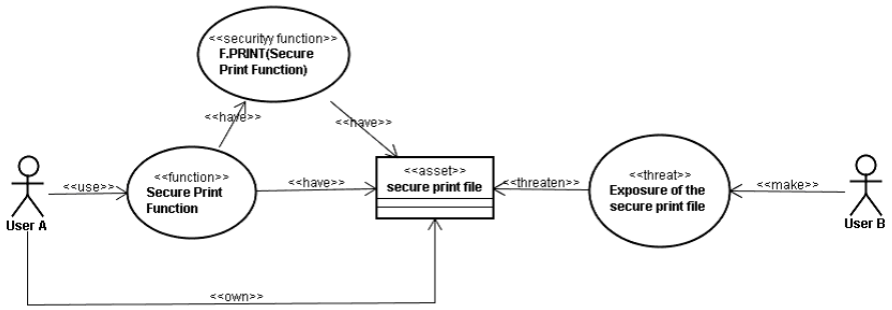


図 10: Security Print Function phase 1

他の利用者に盗み見られる可能性や、他の印刷物に紛れ込む可能性を排除することができる。

- フェーズ 1 この例では security function は function そのものが上記の様にセキュリティ機能を有している為、特にあらたに考察することなく security function として function と同様の名前で定義する。(なお識別のために F.PRINT を付加している)。asset は「画像ファイル」だが識別の為に secure print file とする。また上記説明からセキュリティ文書パスワードというデータも存在することが解るが、ここでは分析を想定して、単純に機能で保護する asset とその threat であるセキュリティ文書の不正入手のみを記載する（図 10)。

- フェーズ 2 図 11 は「セキュリティ文書プリント機能」の例に対してフェーズ 2 までの作業を終了した例である。Security print file に対する不正アクセスはパスワードによるアクセス制御により対策（counter）され、そのパスワード（security print password）の不正入手の攻撃に関しては機能的にパスワード規約が実装されている（もしくは仕様上実装される）ことにより対策している。「セキュリティ文書プリント機能」では通信の盗聴に対する objective が機能的に対応できていない為、盗聴防止のための operational environment を記述する。この際、TOE の設置環境に前提として要求すべき環境条件として定めたほうが適当であると判断した場合には、threat

ではなく前提条件（assumption）として定義する。ST に記述する際は asset により排除される（そもそも想定されない）threat は記述しない。次に、不正なセキュリティ文書パスワードの入手という threat に対して、objective によりパスワード規制を設け対応しているが、例えば TOE 外であるクライアント PC へのパスワード入力が平文であった場合の除き見や、ログインしたまま席を離れた場合のパスワードの盗聴や不正な変更に対応できない。そういった TOE の機能ではそもそも十分に対策できない threat に対して、operational environment を記述し対策する。フェーズ 2 が終了した時点で、全ての threat や policy、及び assumption に対して必ず一つ以上の対策が接続されているはずである。

またこの段階で、想定した Threat に対抗する為に、F.PRINT には以下の機能の実装、及び運用環境での対策が必要であることが確認できる。

- secure print file へのアクセスにはパスワードによる認証が必要であること。
- パスワードは強固な値（桁数、ロック機構、文字種等）しか設定できないこと。
- パスワード入力時に入力文字を隠蔽するアプリケーションを用いること。
- ログイン中の離席時には必ずログアウトする運用を徹底すること。

この情報を元に、開発へのフィードバック、ガイダンスへのフィードバックを行うことも考えられる。

- フェーズ 3 CC パート 2 から、objective を満たすのに必要十分であり、かつセキュリティ文書プリント機能の要件を規定するための SFR を抽出し、記述した例を図 12 に示す。簡単の為に、security function と objective 以外は省略した。それぞれの SFR の意味は表 5 のとおりである。表中の説明は CC パート 2 の概要部分の抜粋である。まず、許可された操作（セキュリティ文書プリント機能）を動作させる前に、利用者がこの機能を許可された人であることを認証することを求める objective に対して、CC パート 2 から、主体の識別と認証に関する SFR を抽出する。また、この例では、認証情報の管理に関する要件と、認証失敗時の要件を追加することにより、不正な操作から認証情報を保護するとともに、認証メカニズムの機能強度を許容される程度に高めることを求めている。

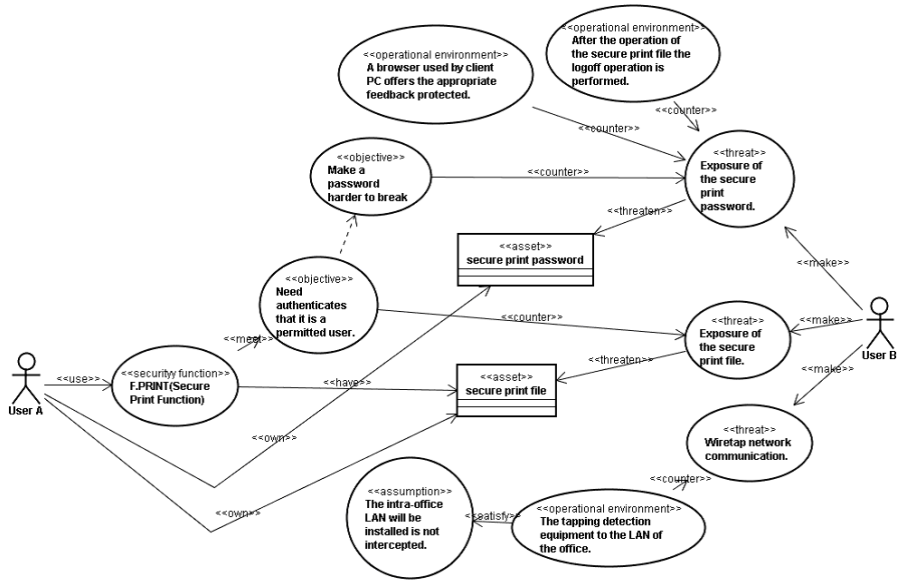


図 11: Security Print Function phase 2

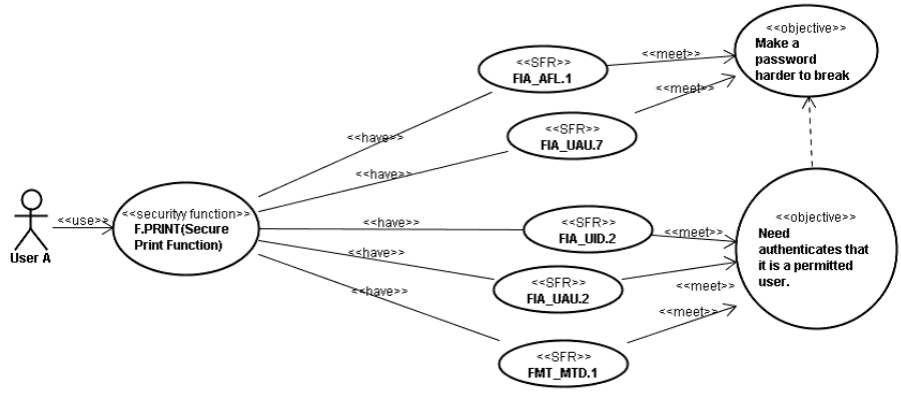


図 12: Security Print Function phase 3

## 4.6 従来方法との比較

前章で提案した分析手順を用いることにより、CC 認証制度に伴う 評価機関による評価に対して、以下のメリットがある。

この比較に用いるチェックポイントは、 CEM と呼ばれる評価者が用いる評価基準の評価すべき項目であり、実際に従来、文章化された ST を評価員が読解し、評価基準を満たしているか評価していたポイントである。ここでは我々の提案方式を用いた場合に、従来方法と比較してどういったメリットがあるかを考察する。

表 5: SFR の説明

| SFR | CC のガイドライン上の説明 |
|---|---|
| FIA_AFL.1 | 認証失敗時の取り扱いは、利用者の不成功の認証試行が特定した数になった後、セション確立プロセスを終了できることを要求する。また、セション確立プロセスの終了後、その試行が行われた利用者アカウントあるいはエントリポイント (例えば、ワークステーション) を、管理者定義の条件になるまで TSF が無効にできることも要求される。 |
| FIA_UAU.7 | 保護された認証フィードバックは、認証の間、限定されたフィードバック情報だけが利用者に提供されることを要求する |
| FMT_MTD.1 | TSF データの管理は、許可利用者が TSF データを管理することを許可する。 |
| FIA_UAU.2 | アクション前の利用者認証は、TSF がその他のアクションを許可する前に、利用者の認証を要求する。 |
| FIA_UID.2 | アクション前の利用者識別は、TSF がその他のアクションを認める前に、利用者が自分自身を識別することを要求する。 |

#### 4.6.1 5.1. フェーズ１及びフェーズ２におけるメリット

CEM に記載されたチェックポイントのうち、フェーズ２までに関係する主なチェック項目のうち以下に関して、提案方式の図により満たされていることを比較的容易に確認できる。開発者の立場に立てば、提案方式に従い作成した図を持ちうることにより、下記ポイントが満たされた ST を作成できると言える。

- TOE 及び／または運用環境によって対抗する必要がある脅威を記述すること
- 全ての脅威は、脅威エージェント、資産、有害なアクションの観点から記述すること
- セキュリティ対策方針（TOE、運用環境）が明確に識別されていること
- TOE のセキュリティ対策方針が脅威、組織のセキュリティ方針までさかのぼれることを確認すること

- 運用環境のセキュリティ対策方針が脅威、組織のセキュリティ対策方針、前提条件までさかのぼれることを確認すること
- 脅威にさかのぼるセキュリティ対策方針（TOE、運用環境）が必要であること
- 組織のセキュリティ対策方針にさかのぼるセキュリティ対策方針（TOE、運用環境）が必要であること
- 前提条件にさかのぼるセキュリティ対策方針（TOE、運用環境）が必要であること

但し、以下の点については、提案方式の図から完全に把握することは既存の方式と同様に困難であり今後の検討課題である。

- 脅威にさかのぼるセキュリティ対策方針（TOE、運用環境）が十分であること。
- 組織のセキュリティ方針にさかのぼるセキュリティ対策方針（TOE、運用環境）が十分であること。
- 前提条件にさかのぼるセキュリティ対策方針（TOE、運用環境）が十分であること。

### 4.6.2 フェーズ 3 におけるメリット

フェーズ 3 においても以下のチェックポイントに対して提案方式はメリットがある。

- 各 SFR が TOE のセキュリティ対策方針にまでさかのぼれること
- どのコンポーネントが SFR を提供するか
- 依存する SFR を提供する、もしくは提供しない場合の理由を明確にすること

但し、フェーズ 1、2 と同様に SFR の十分性を確認するのは図示したとしても完全に確保することは困難であり、今後の検討課題である。なお、フェーズ 3 で追記する SFR の過不足は今までのフェーズ 1〜2 で発生する過不足による発生する再開発や仕様の見直しが発生するケースは少ない。むしろフェーズ 3 の分析によるメリットは文章からの確認が非常に困難である SFR の対応付けが比較的容易に読み取れることである。

さらに評価機関の立場に立った場合、評価の際のチェックポイントに対するメリットがある。つまり、例えば評価機関が評価の際に提案方式の図を確認することにより（図の内容が ST に正直に写されているかの確認は必要となるが）評価作業が効率化され評価に必要な時間が短縮できることが考えられる。

## 5 比較研究

ユースケース図をセキュリティに対して拡張した研究例は多く、我々の提案方式はそれらに大いに依存している。McDermott と Fox による アビューズケース図（Abuse case Diagrams) [6] においては、通常のユースケースに加えて、インタラクションの結果がシステムに対して害があるユースケースをアビューズケースと呼んでいる。

Sindre と Opdahl によるミスユースケース図 [10] においては、ミスユースケース（Misuse case）とミスユーザ（Misuser）というアクターが明示的に導入されている。攻撃者とは意図的にシステムに対して害を為す者という定義に従うと、ミスユーザは、不注意にシステムに害を為す場合が含まれるので、意味としては広義である。Sindre と Opdahl [10] においては、ミスユーザとミスユースケースは、色による区別が行われており、我々のアプローチのようにステレオタイプを用いた拡張方法を取っていない。それに対して Firesmith におけるセキュリティユースケース図 (Security Usecase Diagrams) [2] においては、 Security と Misuse をキーワードとして用いて、通常のユースケースと区別を行っている。我々はこれを修正してステレオタイプとして定義したものを利用している。

UMLsec [3] は UML を基にしたセキュアなシステム開発手法であり、配置図、コラボレーション図、クラス図などの図に関して、様々なセキュリティに関連する特徴を記述するための拡張がなされている。例えば、 配置図に対しては、新たにセキュアな通信経路などを導入することで拡張している。UMLsec においては、ユースケース図は特に大きな拡張が行われていない。主な理由としては、 UMLsec はより設計に近いフェーズでの利用を目指しており、要求分析レベルでの利用についてはまだ弱いことに起因すると考えられる。

アクティビティ図に関するセキュリティを意識した拡張としては Sindre によるものがある [9]。

第 3 著者はセキュリティ要求分析手法に関する講座開発を行った経験があり、そこでは主に Liu による i* に基づいた方法論 [5] が用いられている [11]。

われわれの提案手法は、主にセキュリティに関する特性をいかにモデ

ル化するかを取り扱っており、開発プロセス全体については言及していない。セキュリティの分野において、開発プロセスについて特に注目したものとしては、Mead らによる SQUARE がある [7]。SQUARE は開発プロセスの定義と、各プロセス毎に最善の実践方法（best practice) を提案している点において優れている。しかし、各開発プロセスを見ると、我々が主張する「資産（Asset)」に関する観点が欠けていることが分かる。本技術レポートの第 3 著者は、その点を Mead に指摘したことがあり、彼女は現在、「資産（Asset)」を入れた開発プロセスを SQUARE に導入することを検討しているとのことである。

第 3 著者と第 4 著者はミスユースケース図に資産（Asset）とセキュリティゴールを導入し、それに対するプロセスを提案している [8]。KAOS のようなゴール指向要求分析方法論とミスユースケース図の良い部分を選択した方法論であると言える。プロセスモデルにおいては、システムに関するセキュリティゴールとシステムに関するセキュリティゴールを分け、各々を獲得するためのプロセスを示している。

## 6 結論

本技術レポートでは，ユースケース図を拡張し CC (Common Criteria) との親和性を高めたセキュリティ要求分析方法について考察し，具体的な分析手順を提案した．さらに，実際に公開されている ST に記載された特定の機能に対して，提案方式による分析を行い，ST に記載されるレベルに過不足ない項目が洗い出せることを確認した．提案方式を用いることにより，特定のシステムに対して CC 認証を視野に入れたセキュリティ効率的な分析が可能と言える．これは分析手法を用いず文章による ST を記載した場合に比較して，ST の記載段階，さらには機能追加や CC 認証の評価段階における指摘による ST の修正が発生した際も，4 章で比較したメリットにより改版が効率良く行えると言える．機能や対策の十分性に関しては提案方式によるモデル図から機械的に読み取ることは不可能であるが，それは今後の検討課題である．

## 参考文献

[1] *Common Criteria for Information Technology Security Evaluation Version 3.1, Part1, Part2, Part3.* August 2006.

[2] Donald Firesmith. Security use cases. *Journal of Object Technology*, 2(1):53–64, 2003.

[3] Jan Jürjens. UMLsec: Extending UML for secure systems development. In *Fifth International Conference on The Unified Modeling Language (UML 2002)*, volume 2460 of *LNCS*, pages 412–425. Springer, 2002.

[4] Konica Minolta Business Technologies, Inc. *bizhub 501 / bizhub 421 / bizhub 361 / ineo 501/ ineo 421 / ineo 361 Control Software Version: A0R50Y0-0100-G00-11 (System Controller), A0R50Y0-1D00-G00-10 (BIOS Controller)*, October 2008.

[5] L. Liu, E. Yu, and J. Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting. In *International Conference on Requirements Engineering(RE 2003)*, pages 151–161. IEEE, 2003.

[6] John P. McDermott and Chris Fox. Using abuse case models for security requirements analysis. In *ACSAC*, pages 55–64. IEEE Computer Society, 1999.

[7] Nancy R. Mead and Ted Stehney. Security quality requirements engineering (square) methodology. *ACM SIGSOFT Software Engineering Notes*, 30(4):1–7, 2005.

[8] T. Okubo, K. Taguchi, and N. Yoshioka. Misuse cases + assets + security goals. In *2009 International Conference on Computational Science and Engineering*, pages 424–429, 2009.

[9] Guttorm Sindre. Mal-activity diagrams for capturing attacks on business processes. In Peter Sawyer, Barbara Paech, and Patrick Heymans, editors, *REFSQ*, volume 4542 of *Lecture Notes in Computer Science*, pages 355–366. Springer, 2007.

[10] Guttorm Sindre and Andreas L. Opdahl. Eliciting security requirements with misuse cases. *Requirments Engineering Journal*, 10(1):34–44, 2005.

[11] Kenji Taguchi and Yasuyuki Tahara. Curriculum design and methodologies for security requirements analysis. *Progress in Informatics*, (5):19–34, 2008.