情報処理学会論文誌 Vol.55 No.9 2134–2148 (Sep. 2014)

CC-Case—コモンクライテリア準拠のアシュアランスケースによるセキュリティ要求分析・保証の統合手法

金子 朋子[1,a)] 山本 修一郎[2] 田中 英彦[3]

受付日 2013年11月29日, 採録日 2014年6月17日

概要：ソフトウェアの開発において，顧客の要求を適切に把握し実現させることは非常に大切なことである．しかし上流工程における要求分析が不十分であるためにシステム開発に重大な影響を及ぼすことは多い．システムや製品が望ましい性質を持ち，危険な状況に陥らない保証を顧客から望まれている．そこで本論文では CC-Case と名付けたアシュアランスケース（ISO/IEC15026）とコモンクライテリア（CC: ISO/IEC15408）によるセキュリティ要求分析・保証の統合手法を提案する．CC-Case はセキュリティ要求分析を実施するとともに CC 準拠の保証もでき，脅威に対して保証できる範囲を明確にし，CC に基づくセキュリティ仕様を顧客と合意のうえで決定できる手法である．本論文では CC-Case のセキュリティ要求を獲得する際の技術的な難しさへの対応と CC-Case の保証の意義を考察する．さらに CC-Case の長所の活用による，CC やアシュアランスケース自体に内在する課題解決の可能性を考察する．

キーワード：セキュアシステム開発方法論，セキュリティ要求分析，アシュアランスケース，コモンクライテリア，GSN，ISO15206，ISO15408

CC-Case As an Integrated Method of Security Analysis and Assurance Using Common Criteria-based Assurance Case

Tomoko Kaneko[1,a)] Shuichiro Yamamoto[2] Hidehiko Tanaka[3]

Received: November 29, 2013, Accepted: June 17, 2014

Abstract: It is important to grasp and realize needs of customers in the system development. However, lack of requirements analysis in the earlier phase often gives a crucial influence to the system development. Customers expect that systems and products satisfy the necessary conditions and guarantees not to fall into any dangerous situations. We show the description of countermeasures and procedures which clarify scope of assurance for the menace, and which obtain an agreement on the assurance level with the customer using Assurance Case and Common Criteria though our original method named CC-Case. CC-Case can provide not only security requirement analysis method but also assurance according to the standard of Common Criteria. We consider its correspondence to technical difficulty with the acquisition of security requests, and significance of its assurance. We show how to solve the problems of Common Criteria and Assurance Case by CC-Case' merits.

Keywords: secure system design methodology, security requirement, assurance case, common criteria, GSN, ISO15206, ISO15408

1. はじめに

ソフトウェアのシステム開発において，顧客の要求を適切に把握し，実現させることは非常に大切なことである．ところが，上流工程における要求分析が不十分であるためにシステム開発に重大な影響を及ぼすことは多い．要求分析がうまくいかない理由は，顧客の要望を開発者が仕様

[1] 株式会社 NTT データ
NTT DATA CORPORATION, Koto, Tokyo 135–8671, Japan
[2] 名古屋大学
Nagoya University, Nagoya, Aichi 464–8601, Japan
[3] 情報セキュリティ大学院大学
Institute of Information Security, Yokohama, Kanagawa 221–0835, Japan
[a)] kanekotm@nttdata.co.jp

化する際にギャップが生じるからである．すなわち，顧客（利用者）の早く，安く，良いものを使いやすくといった要望に対して，開発者は利害関係者の合意を図り，IT 技術・方式を決め，要員のスキルやソフトウェアの再利用方法などを定め，要求仕様を作成しなければならない．この顧客要望の要求仕様への変換時にギャップが生じる．

セキュリティ要求分析は，ソフトウェアの一般的機能の要求分析に比べて，顧客と開発者のギャップはさらに大きくなる．セキュリティ要求分析は，分析すべき情報が多様であり，お互いが複雑に関連していることやシステムを取り巻く状況の変化が目まぐるしい中で，新たな攻撃に早く対処する必要があること，セキュリティの実現には，利便性などの他の特性と相反する要求が生じ，バランスを取る必要があるなどの難しい課題を抱えていることがその理由としてあげられる．たとえば，顧客はセキュリティ機能自体に興味がないことが多く，問題が起きないこと，費用がかからないことを漠然と求める．これに対して，開発者は脅威・リスクの洗い出しに漏れはないかが不明確であり，各工程で何をどこまでやればいいのかも不明確であり，新たな脅威への対処は一般には分からないので困難であるといった課題を抱えながら，顧客と何らかの合意をとってセキュリティ仕様を定めていかなければならない．

またセキュリティ要求には，まず脅威の識別，抽出が必要だが，的確に脅威を洗い出したとしても，それに対する対策が不十分では，顧客が望む品質を確保したとはいえない．システムや製品が望ましい性質を持ち，危険な状況に陥らない対策立案と実施の保証を顧客から望まれているのである．

この現状の課題を解決するために，本論文では，コモンクライテリア（CC：Common Criteria. ISO/IEC15408 と同義）[1], [2], [3] とアシュアランスケース（ISO/IEC15026）[4] を用い，セキュリティ仕様を顧客と合意のうえで決定する手法 CC-Case を提案する．2.1 節に後述するようにセキュリティ要求分析には様々な手法がある．しかし，セキュリティが必要になる状況の明確化，特定シーンにおける脅威のモデル化やそれに対する対策立案の手法がほとんどである．本論文では多様な要求に対して，CC の利用により網羅的な要求分析が可能であり，対策の保証も実施する手法を提案している．

本論文の構成は，まず 2 章において関連研究でセキュリティ要求分析手法，コモンクライテリア，アシュアランスケースなどについて述べる．3 章において CC-Case の提案で目的，定義，アシュアランスケースの役割を示す．4 章では検証・妥当性確認のプロセスと適用事例と実用性についてのケーススタディを示し，5 章の考察で CC-Case の利点やセキュリティ保証の意義と CC やアシュアランスケース自体に内在する課題の解決を述べる．6 章で本論文での達成事項と今後の課題をまとめる．

2. 関連研究

2.1 セキュリティ要求分析手法

セキュリティ要求分析では，顧客は要求に基づく機能要求の分析に加えて攻撃者の存在を考慮した非機能要求の分析を必要とする．そこでセキュリティ要求は資産に対する脅威とその対策の記述が必須となる．セキュリティ要求分析のできる手法にミスユースケース [5]，Secure Tropos [6]，i*-Liu 法 [7]，KAOS [8]，Abuse Frames [9] やアクタ関係表に基づくセキュリティ要求分析手法（SARM）[10] などがある．いずれの手法もセキュリティを考慮した脅威分析やそれに対する対策立案・仕様化の手法だが，明示されない非機能要求に関してあらゆる要件をつくすことは難しいのが実情である．セキュリティ要求は非機能要求の 1 つである．非機能要求について，非機能的要求を明示的に表現し，組織的に取り扱うことができる枠組みである NFR フレームワーク [11] が示されている．また発注者と受注者との認識の行き違いや，互いの意図とは異なる理解をしたことに気づかないまま開発が進んでしまう状態の防止を目的としてセキュリティ要求も含めた非機能要求グレード [12] が示されている．これは重要な項目から段階的に詳細化しながら非機能要求の確認を行うツール群である．

SQUARE [13], [14] はセキュリティのシステム品質を高めるために定められた特定の手法によらないプロセスモデルである．SQUARE は生産物の定義に基づいてリスク分析し，セキュリティ要求を抽出・優先順位付け・レビューする手順である．

マイクロソフトのセキュリティ開発ライフサイクル [15] はデータフロー図を詳細化し脅威の観点 STRIDE で脅威分析を実施する．設計による安全性確保を重視し設計段階でセキュリティ要求を抽出している．

しかしながら，セキュリティ要求を抽出・分析・仕様化，妥当性確認，要求管理する全段階をサポートしている要求分析手法もセキュリティ要求分析の標準的な手法もまだできていないのが現状である．

2.2 コモンクライテリアについて

IT セキュリティ評価の国際標準である CC [2] は，開発者が主張するセキュリティ保証の信頼性に関する評価の枠組みを規定したものである [4]．IT 製品（ソフトウェア，ハードウェア）や情報システムを評価・認証する制度として，日本では「IT セキュリティ評価及び認証制度」が運用されている．また CC 相互承認協定により，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．

セキュリティ要求のうち，IT を使って実現する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．CC は，情報技術セキュリティに関連した製

図 1 CC 構成と ST の記載内容

Fig. 1 Composition of CC and the content of ST.

品やシステムの開発者だけでなく，製品を導入・利用する消費者，製品やシステムの評価者などに，有用な規格である．CC は，評価対象（TOE：Target of Evaluation）と運用環境を正確にモデル化し，資産，脅威および対抗策によるセキュリティの概念と関係に基づいて，セキュリティ機能の評価をする．

CC のパート 1 には評価対象のセキュリティ目標（ST: Security Target）やプロテクションプロファイル（PP：Protection Profile）に記載すべき内容が規定されている（**図 1**）．ST は評価対象のシステムが装備するセキュリティ機能を適切に定義し，セキュリティ保証の目標を規定した文書である．ST は，情報セキュリティ評価を行う際には必須となる文書である．また，PP は TOE の個々の種別に対するセキュリティ要求仕様のセットである．

CC のパート 2 に TOE のセキュリティ機能要件（SFR：Security Functional Requirement）が規定されている．セキュリティ機能とは，セキュリティ対策方針つまり，識別された脅威に対抗することを実現するために必要な評価対象のすべてのハードウェア，ソフトウェア，ファームウェア機能の集合である．セキュリティ機能要件はセキュリティ機能の確からしさを検証するために準形式的な言語で記載する．準形式化するために，CC パート 2 には機能要件がカタログ的に列挙されており，選択などの操作にパラメータやリストを特定することにより，準形式的に具体的なセキュリティ機能要件の記載ができる．**図 2** は CC パート 2 の規定 [1] の一部抜粋であり，**図 3** が規定を適用した事例である．具体的な適用の一例を説明すると，図 2 において機能要件の FIA_AFL1.1 が規定されており，「TSF は，[割付：認証事象のリスト]」となっている．そこで図 3 のように「最後に成功した認証以降の各クライアント操作員の認証」，「最後に成功した認証以降の各サーバ管理者の認証」の割付の特定ができる．

CCパート2の規定（一部抜粋）

FIA_AFL.1.1

TSFは、[割付：認証事象のリスト]に関して、[選択：[割付：正の整数値]，「[割付：許容可能な値の範囲]内における管理者設定可能な正の整数値」]回の不成功認証試行が生じたときを検出しなければならない。

図 2 CC パート 2 の規定

Fig. 2 Definition of CC part2.

準形式的な記載事例

[割付：認証事象のリスト]：

・最後に成功した認証以降の各クライアント操作員の認証

・最後に成功した認証以降の各サーバ管理者の認証

[選択：[割付：正の整数値]，「[割付：許容可能な値の範囲]内における管理者設定可能な正の整数値」]：「1～5回内における管理者設定可能な正の整数値」

図 3 準形式的な記載事例

Fig. 3 Exapmple of nearly formal description.

CC のパート 3 にはセキュリティ保証要件（SAR：Security Assurance Requirement）が規定されている．EAL（評価保証レベル）と呼ばれる保証パッケージを定義し，EAL とそのレベルに合わせた実装方法への要求が規定されている．

2.3 アシュアランスケースについて

アシュアランスケース（assurance case）とは，テスト結果や検証結果を証跡としてそれらを根拠にシステムの安全性，信頼性を議論し，システム認証者や利用者などに保証する，あるいは確信させるためのドキュメントである [16]．アシュアランスケースは欧米で普及しているセーフティケース [17] から始まっており，近年，安全性だけでなく，ディペンダビリティやセキュリティにも使われ始めている．その場合，それぞれディペンダビリティケース，セキュリティケースと呼ばれ，アシュアランスケースはこれらを総称した手法である．アシュアランスケースは ISO/IEC15026 や OMG の ARM（Argument Metamodel）[18] と SAEM（Software Assurance Evidence Metamodel）[19] などで標準化がすすめられている．

ISO/IEC15026 part2 では，対象範囲，適合性，利用法，アシュアランスケースの構造と内容，適用成果物などについて規定している．アシュアランスケースの構造と内容に対する最低限の要求は，システムや製品の性質に対する主張（claim），主張に対する系統的な議論（argumentation），この議論を裏付ける証跡（evidence），明示的な前提（explicit assumption）が含まれること，議論の途中で補助的な主張を用いることにより，最上位の主張に対して，証跡や前提を階層的に結び付けることができることである．この定義を満たすものはアシュアランスケースとみなせる．

アシュアランスケースは，解決が必要とされる具体的な表記方法を示すことにより記述が容易になる．代表的な表記方法は，欧州で約 10 年前から使用されている GSN（Goal

Structuring Notation）[20] であり，要求を抽出した後の確認に用い，システムの安全性や正当性を確認することができる．他に法律分野でアシュアランスケースの理論的背景となる Toulmin Structures [21] や要求，議論，証跡のみのシンプルなアシュアランスケースである ASCAD [22] もある．日本国内では GSN を拡張した D-CASE [23], [24] が JST CREST DEOS プロジェクトで開発されている．システムのサービス提供段階においてアシュアランスケースを用いた提案 [25] もなされている．

2.4 セキュリティケースについて

GSN を提唱した Kelly ら [26] が Security Assurance Cases の作成に関する既存の手法とガイダンス，セーフティケースとセキュリティケースの違いなどを述べているが，具体的に作成したセキュリティケースの事例は示していない．Goodenough ら [27] はセキュリティに対するアシュアランスケース（セキュリティケース）作成の意味を説明している．Lipson ら [28] は信頼できるセキュリティケースには保証の証跡こそが重要であると主張している．

Ankrum ら [29] は CC，や ISO154971，RTCA/DO-178B という 3 つの製品を保証するための規格を ASCAD でマップ化し，ASCE などのアシュアランスケースツールが有効であり，保証規格を含むアシュアランスケースは似た構造を持つことを検証している．CC に対しては，PART3 セキュリティ保証要件についてのみの検討を行っている．

Bloomfield らはセキュリティに関するアシュアランスケースの検討状況 [30], [31], [32] を公開している．CC の役割と保証レベルの関係などを検討しているが，CC の規格全体をアシュアランスケースに利用する方法は検討していない．Vivas ら [33] は，システム開発にともなうアシュアランスケースをプラットフォームに統合した保証手法を提唱している．PICOS（コミュニティサービスのプライバシーと識別管理）プラットフォームのもとに開発され，セキュリティのシステム開発のライフサイクルを通じてセキュリティケースを構築，維持するための方法論になっている．ただし，この要求段階においては UML と同等の記法を利用している．また，システムセキュリティリスク対策を確認するためにセキュリティ CC 分解の参照モデル [34], [35], [36] が示されているが，CC のプロセスを用いた要求分析手法としてアシュアランスケースを用いてはいない．他にセキュリティケースを用いた対策立案方法の提案 [37] がなされているが，CC による保証はなされていない．

3. CC-Case の提案

3.1 CC-Case の目的

セキュリティ要求を獲得する際の技術的な難しさに対応することと同時に CC 準拠の保証をすることが CC-Case の目的である．セキュリティ要求を獲得する際の技術的な難しさには ① 扱う情報に対する複雑性，② 状況の変化，③ トレードオフの 3 つの観点があるといわれている [38]．現状のセキュリティ要求分析手法は，特定のシーンにおいての脅威分析やそれに対する対策立案の手法がほとんどであり，上記 3 つの観点に網羅的に適切な対応が可能なセキュリティ要求分析手法はまだ確立されていない．

CC-Case のセキュリティ要求分析はこれらの難しさに対応できることを目指す．さらに，CC-Case はセキュリティ要求分析を実施するとともに CC 準拠の保証も利用できることを目的にしている．

3.2 CC-Case の定義

CC-Case は CC とアシュアランスケースの長所を統合したセキュリティ要求分析手法かつ保証手法である．

本手法ではセキュアな仕様を作成するために，セキュリティコンセプトの定義，対策立案，要約仕様の手順を定め ST に必要な成果物を作成する．この手順をアシュアランスケースとして定義し，証跡を残す．こうして作成したセキュリティ仕様アシュアランスケースは，CC 準拠と顧客と合意による保証の根拠となる．この一連の作業を行う手法が CC-Case である．

図 4 に CC-Case のセキュリティ仕様作成の手順と用いる入力ならびに生成される証跡の関係を示す．セキュリティ仕様のアシュアランスケースは，システム構築時の入力（前提），手順，証跡を含んだ手法である．すなわち，作業は顧客の要求などの前提条件を入力とし，手順に従って進められるが，それとともに生成される証跡によってアシュアランスケースが必要とする情報ができあがっていく．CC 準拠の保証とは，ST を作成するためのもととなる内容をアシュアランスケースの根拠として残すことである．

図 4 は CC-Case 要求分析の手順やライフサイクルでの位置付け，保証の意義を示した全体像である．

なお，本論文における CC-Case の対象範囲は要求段階のすべてのプロセスを含むが，設計段階からサービス提供段階は含まない．また CC-Case の適用対象はシステムまたは製品である．CC-Case は顧客と開発者との合意を形成する手法であるが，製品開発など，仕様を決める際に承認を取る特定の顧客がいない場合は，要件を決めるうえでの関係者と読み替える．

CC-Case は論理モデルと具体モデルの 2 層構造を持つ．論理モデルは論理的にセキュリティ仕様アシュアランスケースを作成するプロセスを提示し，具体モデルは実際の事例を記述する．つまり論理モデルとは，セキュリティコンセプトの定義，対策立案，要約仕様の手順を定めたプロセスのアシュアランスケースである．具体モデルとは，論理モデルの最下層ゴールの下に作成される実際のケースに応じた成果物のアシュアランスケースである．具体モデルは証跡を最下層に提示するまで適宜論理分解されて記述

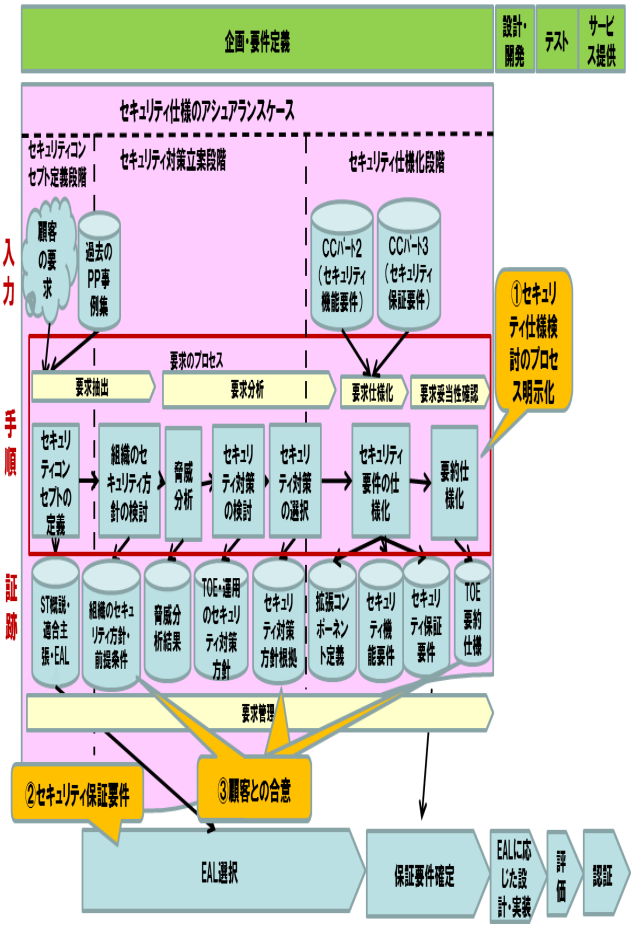

図 4 CC-Case の全体像

Fig. 4 Whole model of the requirement stage of CC-Case.

される．具体モデルは実際のケースにおける ST の証跡と合意による顧客の承認結果を証跡として残す．各種証跡は次々と貯まりその結果，論証に使えるものになる．要望は確定的ではなく，変化することがありうるが，変化に応じた証跡を残すことが必要である．そのため CC-Case では，すべての証跡を要求管理 DB に格納し，変更要求に随時応じられるようにする．具体モデルの各証跡は ST として必要な項目をすべて含むように作成する．セキュリティ要求分析実施プロセスにより，保証のできる証跡を残していき，要求管理として実施される．

図 **5** に論理モデルと具体モデルを図示し，その関係の説明は 3.3 節 (2) で示す．事例を通じた説明は 4.2 節に示す．

3.3 CC-Case におけるアシュアランスケースの役割

(1) CC-Case と GSN

CC-Case はアシュアランスケースの代表的な記法である GSN を使用する．GSN の構成要素を**表 1** に示す．

GSN の構成要素がアシュアランスケースの中でどのように用いられているかを**図 6** で具体的に説明する．CC-Case の最上位のゴールは「CC-Case で作成されたセキュリティ仕様はセキュアである」である．これを最上位のゴールと

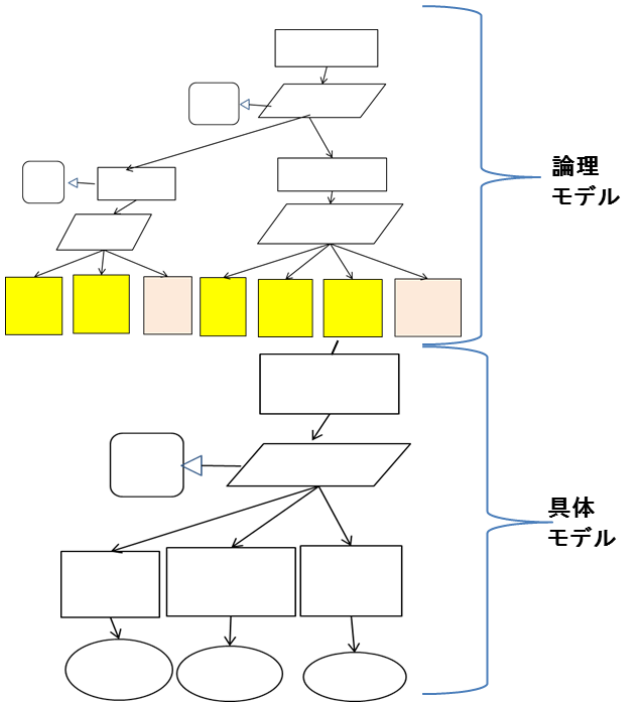

図 5 論理モデルと具体モデル

Fig. 5 Logical model and concrete model.

表 1 GSN の構成要素

Table 1 Contents of GSN.

名称	図式要素	説明
ゴール（主張）		システムが達成すべき性質を示す。下位の主張や説明に分かれる
戦略（説明）		主張の達成を導くために必要となる説明を示す。下位の主張や説明に分解される
コンテキスト（前提）		主張や説明が必要となる理由としての外部情報を示す
未定義要素		まだ具体化できていない主張や説明であることを示す
証跡		主張や説明が達成できることを示す証拠

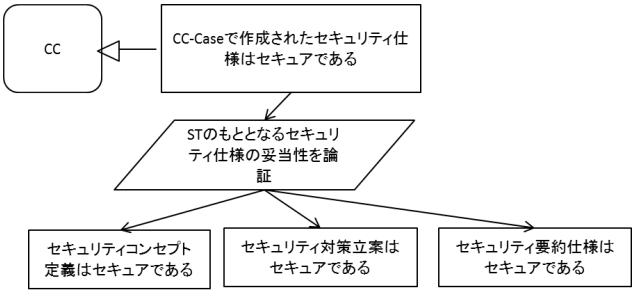

図 6 セキュリティ仕様のアシュアランスケース

Fig. 6 Assurance case of the security specification.

するアシュアランスケースは「CC」をコンテキスト（前提）とし，「ST の元となるセキュリティ仕様の妥当性を論証」する戦略（説明）によって，「セキュリティコンセプト定義はセキュアである」と「セキュリティ対策立案はセキュアである」と「セキュリティ要約仕様はセキュアである」の 3 段階のサブゴールに分かれる．前提とサブゴール

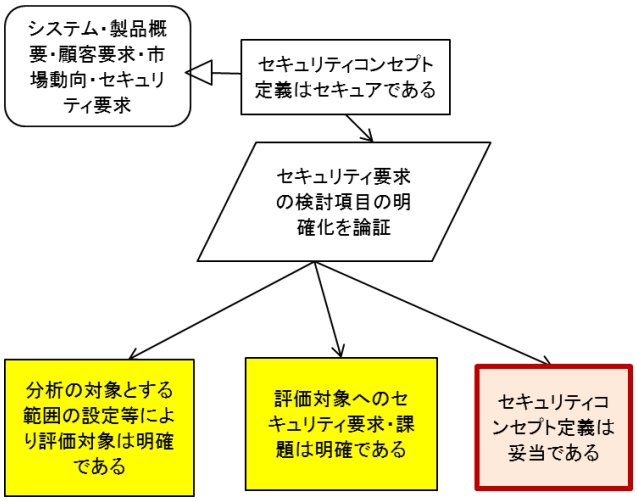

図 7 セキュリティコンセプトの定義段階
Fig. 7 Defining security concept stage.

に分かれる戦略の明示により論理関係を明確にしたうえで，各サブゴールが成り立つことで，最上位のゴールが成り立つことが保証される．

(2) セキュリティ仕様のアシュアランスケース

図 6 はトップのアシュアランスケースであり，そのサブゴールをさらに展開していくことにより具体的な作業が決まっていく．すなわち図 6 の一番左のサブゴールを展開したのが**図 7**，2 番目のサブゴールを展開したのが**図 8**，3 番目のサブゴールを展開したのが**図 9** である．図 6〜図 9 までの定まった作業の手順が論理モデルであり，図 4 の手順を詳細化している．論理モデルの各最下位ゴールの下に実際ケースにおいて，ST の内容の証跡とそれに至るまでの論理関係を示すのが具体モデルである．図 4 の証跡は具体モデルの ST の内容の証跡に相当する．**図 10**，**図 11** は具体モデルの適用事例を示している．これらの手順の生成はアシュアランスケースの考え方で展開されている，したがってそれを守ることによりセキュアな仕様ができあがることを主張することができる．論理モデルを展開した上位ゴールは下位階層にあるものすべてが満たされることが必要であると同時にそれだけで十分であるという展開になっている．これが従来手法の単なるプロセスの流れ図への展開と異なるユニークなポイントであり，手法の完全性を示す．以下，図 6 の 3 つのサブゴールをさらに詳しく展開する．

① セキュリティコンセプト定義段階

この段階のアシュアランスケースを図 7 に示す．セキュリティ要求を抽出し，コストなど顧客のニーズをふまえたセキュリティコンセプト検討をする．この段階は ST ではなく，ST の前段階となるセキュリティコンセプトを作成するプロセスである．セキュリティコンセプトは顧客と開発者で何度も繰り返し検討して決めていくものである．本段階では，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．十分な要求を抽出した後，次にコストなど顧客の重要戦略をふまえたセキュリティ要求検討を実施する．図 7 の黄色のサブゴールでセキュリティ要求抽出の十分性，検討項目を明確にする．そのうえで肌色（太い枠線）のサブゴールで顧客とセキュリティコンセプト定義に対する顧客の合意をとることによる妥当性確認をする．このように CC-Case では合意による保証を妥当性確認としている．

② セキュリティ対策立案段階

この段階（図 8）は，セキュリティコンセプトをもとに評価基準を定め，脅威分析を行い，ならびに，対策の立案と評価，対策の選択をする．そしてそれぞれがセキュアであることを検証する．以上のような作業のステップごとにセキュリティ要求間の対応関係や論理関係の分析と根拠を顧客に提示し合意を得て決定していくがそれを整理して，以下の 3 ステップとした．

ステップ 1：最初に評価対象の範囲を定め，次にどこまでのレベルで保証するのかを EAL として定め，評価基準が妥当であるかを確認し，顧客の承認を得る．

ステップ 2：保護対象とする資産に関する脅威モデルを定義し，それに基づいて脅威を分析し，想定する特定の運用環境における対策を抽出する．脅威と対策の関係の評価が妥当であることを論証し，セキュリティ対策方針根拠に対する顧客の承認を得る．

ステップ 3：ステップ 2 で上がった対策案の中から実施する対策を選択し，対策を実施しないリスクは残存リスクとして管理していく．次にこれらの選択が妥当であることを論証し，セキュリティ対策の選択結果に対する顧客の承認を得る．

この対策立案の 3 ステップは「① 評価基準が妥当である，② 対策評価が妥当である．③ 対策の選択が妥当である．」のシステマティックな手順をきちんとふんでセキュリティ対策を顧客と合意し，証跡を残すことを規定していることになる．なお，アシュアランスケースを作成するうえでの議論分解パターンとしては Bloomfield が示した 7 つの分解パターン [34] とその応用パターン [24] があるが，上記セキュリティ対策立案段階には，応用パターンの 1 つである代替案比較パターンを参考モデルにした．

セキュリティ対策立案段階は，脅威分析やリスク評価機能対策を検討したうえで ST を作成するプロセスに相当する．ただし，セキュリティ対策立案段階では単に ST 項目をそのまま利用しているわけではなく，次のようなポイントを考慮してプロセスを構成した．すなわち ① 対策への評価の妥当性確保のためのアシュアランスケースの類型パターンの応用，② 顧客合意リスクへの対応など，妥当性確認プロセスを 3 ステップともに設定，③ CC 準拠の保証のための ST 項目作成，④ 実施対策の選択プロセスの設定，⑤ 残存リスクへの対応プロセスの設定の 5 つを考慮している．これにより脅威分析結果をもとに適切な対策立案を

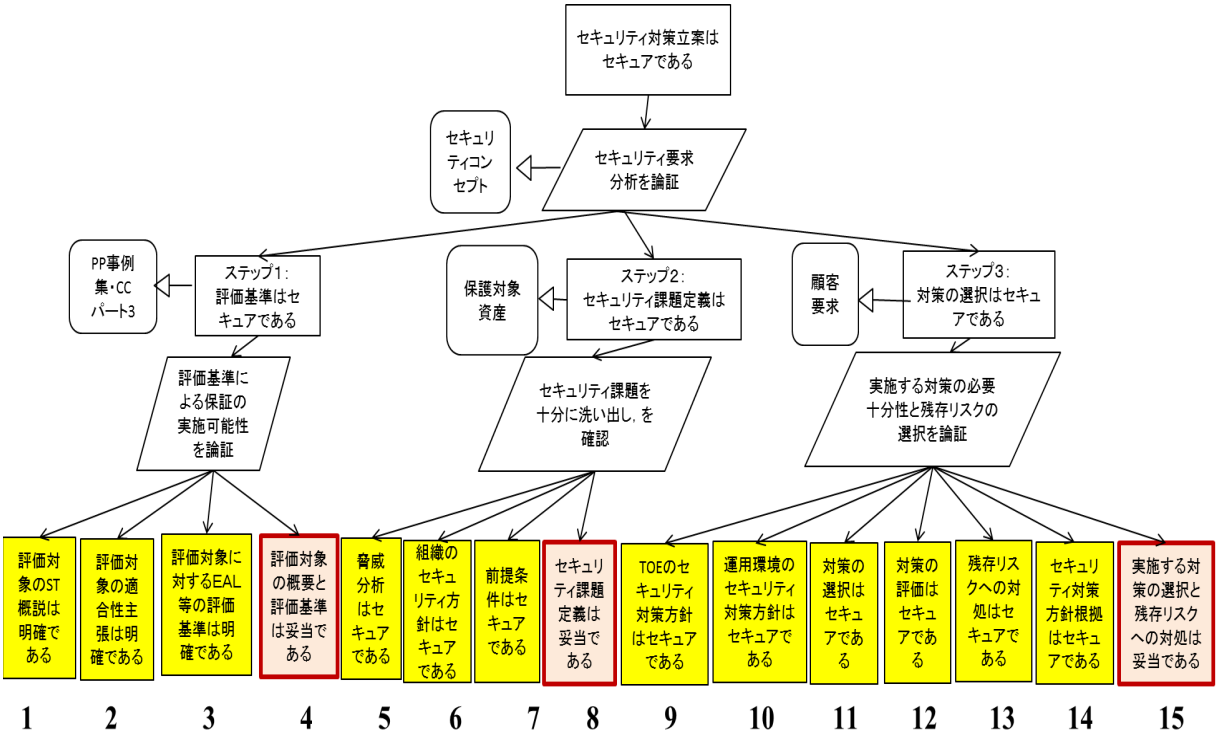

図 8 セキュリティ対策立案段階

Fig. 8 Stage of making security countermeasures.

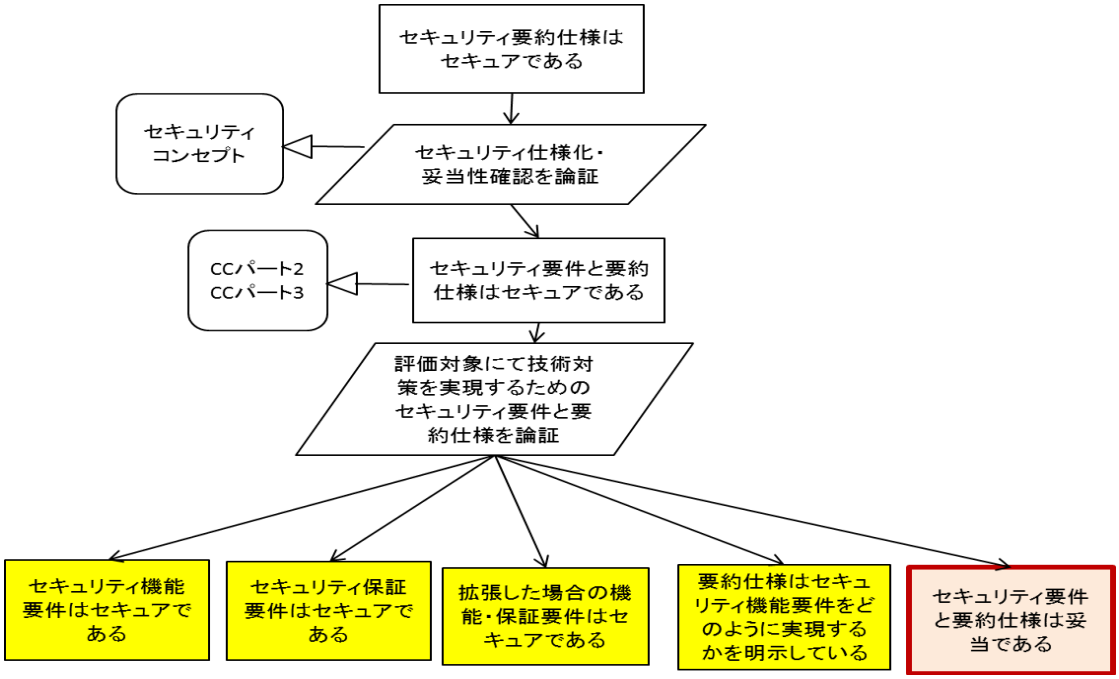

図 9 セキュリティ要約仕様化段階

Fig. 9 Stage of the security specification.

実施するうえで，ST としての結果に表現される前の現実的な検討必要事項を含めたものになっている．

なお，説明をしやすくするため，図 8 の各最下位ゴールの下に項番をふっている．項番 2「評価対象の適合性主張は明確である」では CC-PART1・PP 事例集をもとに ST がコモンクライテリア自体やプロテクションプロファイル（PP）とどのように適合するかを記述する．さらに対応する PP が存在する場合には PP のセキュリティ要件を利用してより効率的に具体モデルを作成できる．

③ セキュリティ要約仕様化段階

この段階（図 9）では，拡張コンポーネント定義，セキュリティ機能要件，セキュリティ保証要件，要約仕様がセ

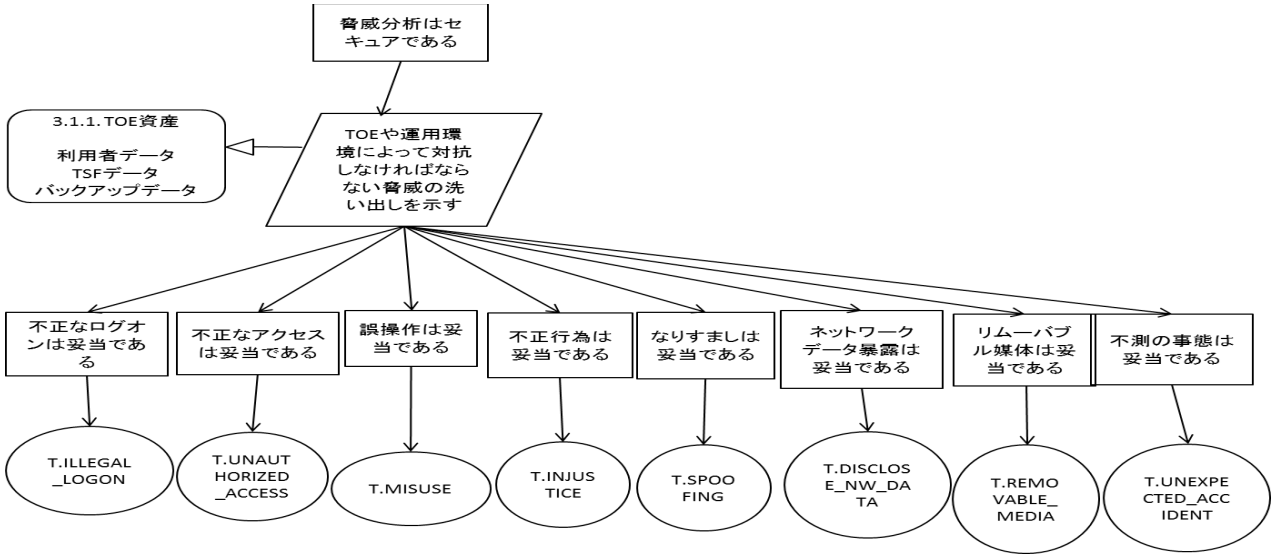

図 10 脅威の適用事例
Fig. 10 Example of the threat analysis.

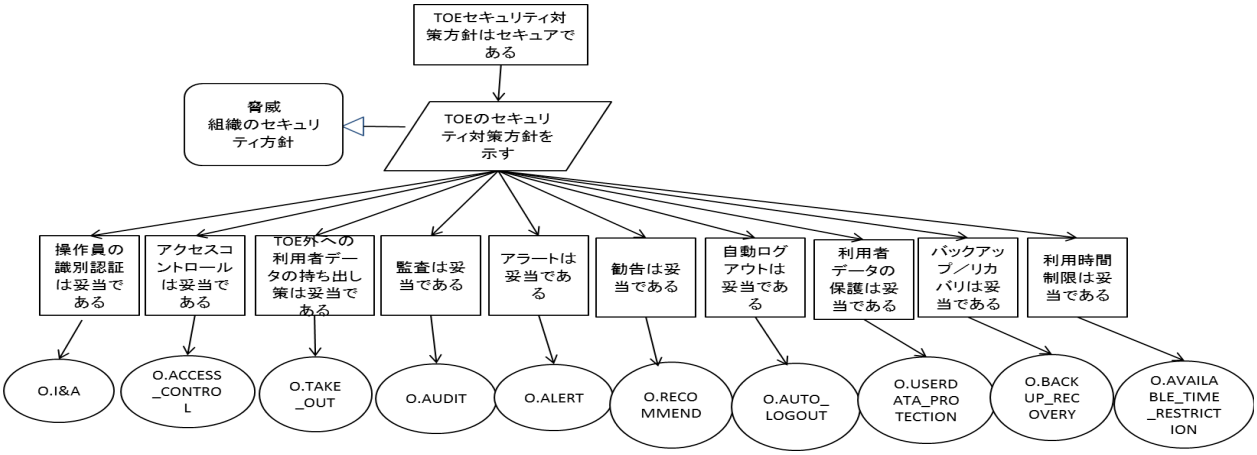

図 11 TOE セキュリティ対策方針適用事例
Fig. 11 Example of the security objectives for the TOE.

キュアであることを検証し，顧客との合意をとる．セキュリティ機能要件はセキュリティ対策立案段階で選択したTOE のセキュリティ対策方針を，技術的な対策として実現するために，CC パート 2 からの機能要件の選択により作成する．セキュリティ保証要件は，CC パート 3 からの保証要件の参照などにより作成する．CC パート 2・3 だけでは機能要件や保証要件を明確にできない場合に，拡張コンポーネント定義し，拡張した機能要件や保証要件として利用する．要約仕様はセキュリティ機能要件を実システム上で実装する方法を示す．このセキュリティ要約仕様に対して顧客の承認を得て妥当性確認をする．

3.4 論理モデルの詳細

前述の図 7〜図 9 までのサブゴールのレベル（最下位ゴール）は具体的モデルを記述する際のトップゴールとなるが，ここでは今一度それらの意味を吟味する．そのために，それらの最下位ゴールごとに目的，プレーヤ，出力に対する確認方法，入力，手続き，出力を規定する．その例としてセキュリティ対策立案段階の最下位ゴール項番 5「脅威分析はセキュアである」のプロセスと項番 9「TOE のセキュリティ対策方針はセキュアである」を以下に示す．

5) 脅威分析はセキュアである
- 目的：TOE が対処しようとする特性やセキュリティの範囲を形式的な作法で定義する（セキュリティ課題定義）ため，評価対象において想定される十分な脅威を引き出し，分析する．
- プレーヤ：開発者
- 出力に対する確認方法：検証
- 入力：保護資産・セキュリティ機能
- 手続き：脅威は資産に対する脅威エージェントの有害なアクションから構成されるため，開発者は保護資産・セキュリティ機能を元に脅威分析を実施，想定される

脅威を洗い出す．脅威エージェント，資産，および有害なアクション．各項目を接頭辞 “T.” の付いた名前で表す．
例：T.ACCESS（アクセス権のない利用者が資源へのアクセスや操作を実行）
- 出力：脅威分析結果

9) TOE のセキュリティ対策方針はセキュアである
- 目的：セキュリティ課題定義を技術的に解決するセキュリティ機能を提供するために TOE のセキュリティ対策方針を規定する．なお，「対策方針はセキュアである」とは方針に従って十分な対策を施すことを示す．
- プレーヤ：開発者
- 出力に対する確認方法：検証
- 入力：セキュリティ課題定義
- 手続き：TOE の技術的なセキュリティ対策方針を自然言語で規定し，TOE のセキュリティ対策方針（Security objectives）を作成する．セキュリティ課題に対する，抽象度の高い解決策を，過度の詳細を省いた，簡明かつ明確な文章で記述する．たとえば，内部対策は通常のレベルでよいが外部からの侵入対策は十分に施すなど．各項目を接頭辞 “O.” の付いた名前で表す．
- 出力：TOE のセキュリティ対策方針

4. ケーススタディ

4.1 検証・妥当性確認のプロセス

表 2 に「セキュリティ仕様のアシュアランスケース」として作成された論理モデルの規模を示す．3 段階の工程に対して，分類は 4，保証条件数は 23 である．どの段階のゴールも検証と妥当性確認を実施している．なお，具体モデルは実際のケースに応じて記述要素数が違うため，この表には記載していない．このように CC 準拠のセキュリティ要求分析とアシュアランスケースによる検証・妥当性確認のプロセスの双方を実施することにより，CC-Case はセキュリティ要求分析手法かつ保証手法といえる．また CC-Case は PP やセキュリティ機能要件のカタログを利用でき，初めからすべてを検討するよりも効率的である．CC-Case が従来の CC の手順と比べて増えているのは顧客の承認による妥当性確認のみである．表 2 の対応するプロセスで妥当性確認は合計 5 つであるが，これはコミュニケーションギャップによる手戻りを起こさないために必要なプロセスである．

表 2 論理モデルの規模
Table 2 Scale of logical model.

工程	分類（最下層戦略数）	対応するプロセス	保証条件数（最下層ゴール数）
セキュリティコンセプト定義段階	1	要求抽出と検証 妥当性確認	3 1
セキュリティ対策立案段階	3	要求分析と検証 妥当性確認	12 3
セキュリティ要約仕様化段階	1	要求仕様化と検証 妥当性確認	3 1
合計	4		23

4.2 具体モデルの事例

CC-Case は具体モデルで実際のケースを記述する．具体的な例示に関しては，IPA の ST 事例 [39] に対し，CC-Case をもとに記述した．その結果，事例全体をアシュアランスケースで書くことができることを確認している．

2.4 節の文献 [34], [35], [36] で示したようにセキュリティケースには共通性があるにもかかわらず，プロセスが明確になっていない．このため個別に作成されたセキュリティケースでは品質のバラツキや作成効率が低いという実用性の問題がある．それに対して CC-Case は論理モデルとして共通性を明確化するとともに，具体モデルを追加することで自然にセキュリティケースを作成できる実用性があり，その結果を証跡として残せる長所も持つ．

図 10 脅威分析は，図 8 セキュリティ対策立案段階における項番 5「脅威分析はセキュアである」を事例化した具体モデルである．保護対象資産は利用者データ，TSF データ，バックアップデータが相当し，不正なログオン，不正なアクセス，誤操作，不正行為，なりすまし，ネットワークデータ暴露，リムーバブル媒体，不測の事態の脅威の洗い出しが妥当であることを検証し，記述したものである．CC-Case を利用する中で得られた知識を資産化することによりカタログ化した脅威のパターンを利用すると不正ログオンなどの脅威の洗い出しが容易になる．各々の検証結果は T.ILLEGAL_LOGON, T.UNAUTHORIZED_ACCESS などの脅威分析の証跡として示される．図 10 は CC パート 1 の付属書 A.6.2 脅威の記載事例に相当し，ST の仕様にのっとった証跡の検証手段を含んでいる．図 10 の一番左の T.ILLEGAL_LOGON の証跡には「T.ILLEGAL_LOGON（不正なログオン）攻撃者が，TOE の正当な利用者になりすまして TOE を利用することにより，利用者データを破壊・改ざん・暴露するかもしれない」と記述される．

図 11 はセキュリティ対策立案段階における項番 9「TOE セキュリティ対策方針はセキュアである」を事例化した具体モデルである．脅威と組織のセキュリティ方針を前提として，操作員の識別認証，アクセスコントロール，TOE 外への利用データの持ち出し策などが妥当であることを検証し，記述したものである．脅威の対応策や組織方針に対して TOE が技術的に実現すべき対策を示している．図 11 は CC パート 1 の付属書 A.7.2.1 TOE のセキュリティ対策方針の記載事例に相当し，ST の仕様にのっとった証跡の検証手段を含んでいる．対策方針化の際は，参考事例としてセキュリティ機能要件に基づいた PP などの豊富なパ

ターン [40] をもとに対策が必要十分であるかを確認する．ここでいう検証とはこの確認を指すが，さらなる検証方法については今後の検討課題である．

図 11 の一番左の O.I&A の証跡には「O.I&A（操作員の識別認証）TOE は，操作員が TOE を利用するときは必ず識別認証されることを保証し，指定された回数以内に識別認証に成功した操作員のみ TOE の利用を許可しなければならない」と記述される．

5. 考察

本章では，CC-Case の特長は何であり，CC-Case の目的はどのように達成できるのかについてや CC-Case の特長を生かすことにより，CC やアシュアランスケース自体に内在する問題点を克服できる可能性を議論する．

5.1 CC-Case の特長

(1) 要求分析と保証の手法

CC-Case は，CC にのっとり要求分析の段階で顧客と合意した範囲におけるセキュリティ保証要件を定め，保証をするために必要なセキュリティ機能要件を，CC に準拠して網羅的に抽出・分析・妥当性検証・仕様化・管理を行う要求分析手法である．さらに CC-Case は，図 4 に示している ① セキュリティ仕様検討プロセスの明示化，② セキュリティ保証要件，③ 顧客との合意のプロセスの明示化による保証手法である．① と ② は CC 準拠による保証である．セキュリティ要求分析の手法は数多いが，CC 準拠と顧客合意の保証をともなう手法は皆無であり，CC-Case の特長である．

以下にプロセスと成果物の観点から，CC-Case はセキュリティ要求分析手法かつ保証手法であることを論述する．

CC-Case のセキュリティ要求分析実施プロセスは「セキュリティ仕様のアシュアランスケース」として ST を作成する際に必要なプロセスが明示化されている．したがって CC に準拠したシステム・製品の仕様要求，セキュリティ要求の対応関係や論理構造を CC-Case を利用しない場合より網羅的に分析しやすい．

IEEE Std1012-2004 [39] によれば，検証（verification）とは，ソフトウェア品質保証のための確認プロセスには開発活動による生産物が開発活動に対する要求に適合していることを判定する活動であり，妥当性確認（Validation）とは，開発されたソフトウェアが意図された利用法とユーザニーズに適合していることを判定する活動である．

CC-Case のセキュリティ要求分析実施プロセスでは ST を作成する際の要求に適合していることを検証する．検証の実施個所は図 7〜図 9 の黄色のゴールである．セキュリティ保証の観点から検証は正確性，完全性，有効性が検証される．正確性とは内容に誤りがないこと，あいまいさがなく明確であることである．CC-Case のゴールは CC 準拠であること，すなわち各証跡が ST として誤りがないことを検証している．また完全性とは漏れがないこと，余分なことがないことである．CC-Case はセキュリティ対策方針根拠，セキュリティ機能要件根拠やセキュリティ機能要件とセキュリティ機能要約仕様の対応関係において要件の必要十分性の検証を実施する．また，有効性とはセキュリティ機能が目的どおりに動作することである．各ゴールに示された「セキュアである」つまり安全であることが保証されることである．

CC-Case ではセキュリティ規格に適合した CC に基づく PP のセキュリティ要件を利用して対象とするシステムのセキュリティ保証を確保する．PP は公開されており，評価対象の種別に対して適用すべきセキュリティ仕様として適切な PP があれば，それを利用することができる．適切な PP が存在した場合はそのセキュリティ要件を利用する．

新しいシステムなどで適切な PP を特定することが難しい場合は，まずは類型化した PP を作成し，同様の PP を知識資産化していくことが重要になる．その知識資産化にはプロセス定義を詳細化したうえでのアシュアランスケースの利用が適している．詳細なプロセス定義で同種のプロセスをカテゴリ化し，アシュアランスケースの証跡として残していくことで再利用がしやすいからである．CC-Case はこのプロセスの詳細化と具体モデルとしての証跡の残し方を明確に定義している．

日本では 2001 年より ISO/IEC 15408 に基づく「情報セキュリティ評価認証体制」[40] を運用しているが，PP の普及は不十分であった．しかし「政府機関の情報セキュリティ対策のための統一管理基準（平成 26 年度版）」において，適切な情報セキュリティ対策の確保のため，「機器等を調達する場合には，「IT 製品の調達におけるセキュリティ要件リスト」を参照し，利用環境における脅威を分析したうえで，当該機器等に存在する情報セキュリティ上の脅威に対抗するためのセキュリティ要件を策定すること」が求められている．また経済産業省による「IT 製品の調達におけるセキュリティ要件リスト」[40] が策定された．本リスト CC に基づいたセキュリティ仕様が製品分野ごとに PP で定義されている．「IT セキュリティ評価及び認証制度」を運営している独立行政法人情報処理推進機構（IPA）では，今後製品分野ごとの PP を拡充していくことを目指している [41]．今後は情報系システムの大規模な SI 現場において利用可能な PP も含め，PP の利用促進が活発化していくことが期待される．また世界では多くの製品・システムに PP が存在する [42]．また CC の業界での最近の動向では，これまで PP の開発は主としてセキュリティ製品の認証を望んでいる企業が行ってきたが，現在，企業が連携して共同 PP（cPP: Collaborative PP）を実現しようとしており [43]，バンキングシステムの PP を共同開発事例などが報告されている [44]．このように最近の CC 状況を

考慮すると，今後本研究の有効性が高まると考える．

また，CC-Case では各段階で示された証跡がユーザニーズに適合していることを顧客と合意することにより，妥当性確認を行う．なお，証跡は具体モデルの最下位で実際の事例に基づいた根拠を示している．顧客との合意プロセスは図 7〜図 9 の肌色（太い枠線）のゴールである．

(2) 関係者の観点で見た利点

開発者・保守運用者にとっては，① ST を作成する際に必要なプロセスが明示化され，製品やシステムの ST が妥当であることを検証できる．② 準形式的なセキュリティ機能要件を利用できるので齟齬が生じず，設計しやすさが期待できる．③ 証跡を要求管理 DB で管理するので，新たな脅威の発生などにともなう仕様変更の対応しやすさが期待できる．④ CC パート 2 に規定されたセキュリティ機能要件は形式化・カタログ化されているので要件の再利用しやすさが期待できる．

顧客（要件決定の関係者）にとっては ① 妥当性確認のタイミング（肌色・太い枠線）が定められており，段階に応じて，要件が顧客ニーズから見て妥当であるかの判断の根拠が与えられる．② コスト・難易度などセキュリティ以外の他のニーズでもどこまでの保証レベルとするかの主張ができる．③ EAL を用いることで，評価保証レベルを明確にし，開発や構成，管理などに関わる個々の保証要件を規定するのではなく，目標とする評価保証レベルだけを指定することができる．必要な保証レベルに応じて製品やシステムを，適正なコストで保証することができる．

5.2 セキュリティ要求を獲得する際の技術的な難しさ

「セキュリティ要求を獲得する際の技術的な難しさ」の 1 つである ① 扱う情報に対する複雑性とは，セキュリティに関する関心事は多く，互いに複雑に関連していることから来る問題である．具体的には分析すべき情報，資産，脅威，機能要件，根拠などセキュリティに関する関心ごとが多いこと，対抗策の妥当性，正当性，機能要件の完全性などの関心ごとが互いに複雑に関連していること，脅威の範囲，前提条件などの開発範囲が不明確であることが該当する．2.1 節にあげたミスユースケース [5] や Abuse Frames [9] などのユースケースのセキュリティ拡張手法や Secure Tropos [6]，i*-Liu 法 [7]，NFR フレームワーク [11] などのゴール指向要求分析方法論は，この複雑な関心事を分析する手順があいまいで特定のシーンにおける部分的な分析をする傾向がある．また手順があいまいなため，検討漏れが生じやすい．これに対して CC-Case はセキュリティ要求分析で扱う脅威，リスク，対策，資産などの複雑な情報を CC にのっとり，より明確に具体的に要求分析手順化することで図 4 の「① セキュリティ仕様検討のプロセス明示化」を実施している．そのため保証に基づいた系統的な分析ができ，検討漏れが発生しにくい．また要求分析をしながら証跡を残していくため論理の完全性を確保できる長所を持つ．

図 7〜図 9 までのサブゴールのレベルで，3 章に論理モデルの詳細なプロセス，4 章に具体モデルの一例を提示した．CC-Case ではセキュリティ要求定義に必要な項目を 23 の最下位ゴール数に規定し，各ゴールに対し，詳細なプロセス定義がある．単純に何の規定にも基づかずセキュリティ要求分析を実施すれば，セキュリティゴールを決めて，思いつく範囲の脅威分析を実施して，それらに対して対策をたてるだけの 3 つ程度のプロセスになりかねない．これに比べると，CC-Case は実施すべき事項が詳細であり，明確である．さらに 1 つ 1 つのプロセスを実施すること自体は効率的で容易化されている．これは PP の適用などによる効率化が可能となっているためである．CC-Case では 3 章に一例を示したプロセス定義を図 7〜図 9 に示すすべてのサブゴールにおいて，実施済みである．また CC-Case では IPA（独立行政法人情報処理推進機構）の ST 事例 [39] 全体に対し，セキュリティ仕様のアシュアランスケースの具体モデルで書くことができ，現実的に利用可能である．このような詳細なセキュリティ仕様検討のプロセス明示化により，CC-Case は “セキュリティ要求を獲得する際の技術的な難しさに対応する” ことができる．

② 状況の変化とは，見えない敵が存在し想定外の新たな脅威が発生することであり，それに対してさらなる対策を繰り返す必要が生じる．CC-Case は，要求管理 DB を設定しており，すべての証跡と論理的根拠を残し，アシュアランスケースの利用により，最上位の主張に対して，階層的に結び付けることができる論理的証拠の提示が可能である．これにより当初の想定とはまったく違う要求が生じたとき，どの機能にどのような影響があるのか，どのように変更すべきかの判断がしやすく，繰り返される変更に対処しやすい．次に ③ セキュリティ要求を考えるときに，競合する要求との間にコストや機能の使いやすさなど，他の要求とのトレードオフが生じる．CC-Case はセキュリティ対策の競合する要件を考慮し，顧客と合意のうえでどこまでの対策を実施するのかを選択するプロセスを明示して顧客が選択可能にしている．これが図 4 の「顧客との合意プロセスの明示」である．妥当性確認は顧客との合意という根拠を持ってトレードオフに対処する．なお，対策立案・選択の枠組みを提示するのがこの論文の主旨であり，選択に必要な条件を提示しどのように選択を行うかについての具体的な手法は今後の研究課題である．これは 3.4 節に前述したように，図 7〜図 9 の肌色（太い枠線）のゴールでの妥当性確認が相当する．妥当性確認は顧客との合意という根拠を持ってトレードオフに対処する．

5.3 CC 準拠の保証

セキュリティ保証を体系的に実施するためには，評価

の枠組みが必要である．これを規定しているITセキュリティ評価の代表的な国際標準はCCである．しかし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[43]．この具体的な手順を与えることによりCC-CaseはSTを作成するための元となる要求分析や脅威分析を実施し，CCに基づく保証を可能とする．

評価対象TOEの中で合理性に矛盾がなければセキュアでなくてもCCに準拠できると考える方がいるかもしれない．しかし最上位のゴールは「CC-Caseで作成されたセキュリティ仕様はセキュアである」を満たすためには，下位のすべてのゴールの達成が必要である．下位ゴールでは，TOEと運用環境を正確にモデル化し，資産，脅威および対抗策によるセキュリティの概念と関係に基づいて，セキュリティ機能が規定され，セキュアであることを検証される．具体的には想定される脅威に対抗するために適切なセキュリティ対策が策定されていること，セキュリティ対策の実現のために適切なセキュリティ機能要件が選択されていること，選択されたセキュリティ機能要件に対して適切なパラメータの割付などが指定されていること，想定される脅威や期待される確からしさに対応した保証要件が選択されていることなどの条件を検証する．これらの条件を満たせばそのため「CCに準拠すること」のゴールは「セキュアであること」を満たすことになる．

なお，代表的な分析手法であるSQUARE[13], [14]，セキュリティ開発ライフサイクル[15]も含め，2.1節であげた従来のセキュリティ要求分析手法には要求分析をしながら，CCのセキュリティ基準にのっとった保証を可能とする手法はなく，この点でCC-Caseにはオリジナリティおよび手法の特長がある．

5.4 CCの課題解決

「コモンクライテリアにおけるセキュリティ要求の規定」[43]には，後述の①から⑤の課題が示されている．それに対してCC-Caseがどのような課題解決の手段を持つかを示す．なお，CC認証製品の維持・機能拡張段階，モデル展開段階は本論文の対象段階ではないが，ライフサイクルサポートを今後の検討課題にしており，当該段階での課題解決手段についても検討する．

① CC導入段階において，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．通常，セキュリティ要件について，市場要求をふまえ，セキュリティ要件の取捨選択などの現実的な調整が必要である．しかし，実際には開発者はシステム・製品の仕様要求，セキュリティ要求，および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・共有する実用的な手段が存在せず，これがST内部の不整合，STと開発エビデンスとの不整合の原因となる[43]．この課題に対して，CC-Caseはシステム・製品の仕様要求，セキュリティ要求の対応関係や論理構造を分析する手段を提供し，見える化を実現している．また市場要求をふまえ，セキュリティ要件の取捨選択などの現実的な調整を顧客との合意のうえで実施することができる．さらに要求管理DBを導入することは開発の管理・証跡の共有の実用的な手段となる．

② CC認証製品の維持・機能拡張段階において，変更要求にともなう対応に漏れがないことが必要だが，自然言語で書かれたST更新ワークのみでは，分析ツールとしての能力が低く，不完全な修正になる可能性が高い．また，ライフサイクルで一貫した要求管理が不十分であり，脅威事象や個別要求の変化への対応が不完全になりやすい[43]．この課題に対して，CC-Caseは自然言語で書かれたST更新ワークよりも，アシュアランスケースの記法で全体が構造化されて一覧に図示されていることや論理モデルでプロセスが規定され，具体モデルのみを更新すればよいため，修正箇所の特定が早く，変更要求にともなう対応漏れを低減させることが期待できる．さらにまた，証跡は準形式的なCCの記法で表記されているので，パターン化して修正箇所の特定をすることもできる．さらにCC-Caseの要求管理DBを作成することにより，ライフサイクルで一貫した要求管理が期待できる．

③ CC認証製品のモデル展開段階においては，部分再利用時のアーキテクチャ上の整合確保が不十分であり，分析漏れによる脅威や脆弱性が残存する可能性がある[43]．この課題に対して，CC-Caseは要求管理DBにCC認証製品のアーキテクチャを具体モデルとしてモジュール化して管理することにより，部分再利用時のアーキテクチャ上の整合確保がしやすくなり，再利用を容易化でき，分析漏れによる脅威や脆弱性を低減可能であろう．

④ 要求工学的なアプローチを実務で生かすためにはCCに最適化した要求記述・分析の方法論やサポートツールが未整備である[43]．この課題に対して，CC-Caseはアシュアランスケースとゴール指向要求工学の統合アプローチを補完的に組み合わせており，CCに最適化した要求記述・分析の方法論を目指している．

⑤ 昨今の製品開発は高度に専門分化され，要求獲得・記述が開発者視点で行われる傾向があり，利用者受け入れ可能な要求獲得が不十分なケースが存在する[43]．この課題に対して，CC-Caseは顧客（利用者）と検討・合意するプロセスを規定しており，受け入れ可能な要求獲得が明示的に実施され，それが不十分なケースを低減できよう．

5.5 アシュアランスケースの課題解決

「主張と証拠」[34]によるとアシュアランスケースには次のように①から④の課題があるが，作成基準のないアシュアランスケースに比べ，CC-Caseではどのように課題解決できるかを以下に考察する．

① 開発方式に関して，アシュアランスケースは一般に，

作成プロセスを開発することにより管理を容易化する必要がある [33]．これに対して，CC-Case は論理モデルとしてセキュリティ仕様検討のプロセスを明示化しており，個別の管理が必要なのは具体モデルのみである．さらに具体モデルの要件は上位の論理モデルによって限定される．したがって管理の容易化や品質向上につながることが期待される．

② 再利用に関して，アシュアランスケースはパターン化による再利用の容易化を課題としている [34]．Ankrum [29] もアシュアランスケースはプロジェクト固有な部分を持ち，再利用可能箇所を区別するのは難しいという課題をあげている．これに対して，CC-Case は論理モデルとして共通性を明確化しているため，論理モデル部分は再利用が可能である．さらに CC-Case はセキュリティ要件を準形式的言語化するので証跡の内容も再利用の容易化が期待できる．

③ 生産性に関して，アシュアランスケースをモジュール化することで作成を効率化する必要があり [34]，Ankrum もアシュアランスケース構築に時間とコストがかかるとしている [29]．これに対して CC-Case は CC に基づいた手順は論理モデルとして共通化しており作成不要である．また収集すべき証跡も規定されているため，これらの規定のないアシュアランスケース構築に比べて，生産性向上が期待される．

④ 確信性に関して，アシュアランスケースを構成する下位のサブゴールや戦略ならびに証拠の成立確率に基づいて，アシュアランスケースの充足性を推論する方法を確立する必要があるといわれている [34]．これに対して CC-Case は CC 基準をベースにして，サブゴールや戦略ならびに証拠を作成しているが，その成功確率に関する検討はまだ行っていない．確信性に関しては今後の検討事項である．

6. おわりに

本論文では，CC とアシュアランスケースの長所を統合したセキュリティ要求分析手法かつ保証手法として CC-Case を提案した．CC-Case の目的，定義，CC-Case におけるアシュアランスケースの役割，特長を示した．さらにケーススタディのうえでケーススタディにより手法の妥当性，実用性の事例を示し，セキュリティ要求を獲得する際の技術的な難しさへの対応や CC 準拠の保証ができることと，CC やアシュアランスケース自体に内在する問題点を克服できる可能性について考察した．

5 章の考察はいずれも定性的な評価であり，以下の点を今後の課題とする．

(1) CC-Case 適用の効果はシステム開発からサービス提供のライフサイクルにわたって利用することで効果を発揮するものであり，ライフサイクルにわたるアシュアランスケースの作成手順と実例を適用した定量的な効果測定が必要である．

(2) CC はその複雑さからデジタル複合機などの特定のセキュリティ機能のハードウェア製品や政府調達対象案件には利用されているものの，それ以外のセキュリティ要求分析にはあまり利用されていないという課題も抱えている．この課題解決のため，CC-Case は認証を取るための利用目的以外においても，一般のセキュリティ要求分析に利用できることを示す．

(3) 情報技術セキュリティ評価のためのコモンクライテリア（CC）と対をなす文書として情報技術セキュリティ評価のための共通方法（CEM）[45] がある．CEM は，評価者によって実施される CC で定義された基準および評価証拠を使用した CC 評価を行うための最低限のアクションを定義している．CC-Case はセキュリティ要求分析とともに保証を実施するための開発手法であり，評価方法である CEM と利用目的は異なるが，CC-Case のプロセス定義において参考にすべき点が多いと考えられる．相互の関係についての整理などは今後の課題である．

謝辞 数々の有益なご指摘を賜った査読者の方々をはじめ，本論文の作成にご協力いただいた情報セキュリティ大学院大学の先生方，田中研究室の皆様や，この研究をするにあたりサポートしていただいた（株）NTT データの皆様，励ましをいただいた恩師，友人，家族に謹んで感謝の意を表する．

参考文献

[1] Common Criteria for Information Technology Security Evaluation, available from ⟨http://www.commoncriteriaportal.org/cc/⟩.
[2] セキュリティ評価基準（CC/CEM），入手先 ⟨http://www.ipa.go.jp/security/jisec/cc/index.html⟩.
[3] 田淵治樹：国際規格による情報セキュリティの保証手法，日科技連 (2007).
[4] ISO/IEC15026-2-2011, Systems and Software engineering-Part2: Assurance case (2011).
[5] Sindre, G. and Opdahl, L.A.: Eliciting security requirements with misuse cases, *Requirements Engineering*, Vol.10, No.1, pp.34–44 (2005).
[6] Mouratidis, H.: Secure Tropos homepage, available from ⟨http://www.securetropos.org/⟩.
[7] Liu, L., Yu, E. and Mylopolos, J.: Security and Privacy Requirements Analysis within a Social Setting, *Proc. IEEE International Conference on Requirements Engineering RE 2003*, pp.151–161 (2003).
[8] Letier, E.: Reasoning about Agents in Goal-Oriented Requirements Engineering, Université Catholique de Louvain (2001).
[9] Lin, L., Nuseibeh, B., Ince, D., et al.: Introducing Abuse Frames for Analyzing Security Requirements, *Proc. IEEE International Conference on Requirements Engineering RE 2003*, pp.371–372 (2003).
[10] 金子朋子，山本修一郎，田中英彦：アクタ関係表に基づくセキュリティ要求分析手法（SARM）を用いたスパイラルレビューの提案，情報処理学会論文誌，Vol.52, No.9 (2011).

[11] Chung, L., Nixon, B., Yu, E., et al.: Non-Functional Requirements In Software Engineering, Academic Publishers (1999).
[12] 非機能要求グレード，入手先 ⟨http://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html⟩.
[13] Mead, N.R., Hough, E. and Stehney, T.: Sequrity Qualty Reqirements Engineering (SQUARE) Methodology (CMU/SEI-2005-TR-009), available from ⟨www.sei.cmu.edu/publications/documents/05.reports/05tr009.html⟩.
[14] Mead, N.R.，吉岡信和：SQUARE ではじめるセキュリティ要求工学，情報処理，Vol.50, No.3 (2009).
[15] Steve, L., Michael, H.：信頼できるコンピューティングのセキュリティ開発ライフサイクル (2005)，入手先 ⟨http://msdn.microsoft.com/ja-jp/library/ms995349.aspx⟩.
[16] 松野 裕，高井利憲，山本修一郎：D-Case 入門—ディペンダビリティ・ケースを書いてみよう！，ダイテックホールディング (2012).
[17] Tim, K. and McDermid, J.A.: Safety Case Construction and Reuse using Patterns, *Proc. 16th International Conference on Computer Safety, Reliability and Security* (*SAFECOMP'97*), Springer-Verlag (1997).
[18] OMG: ARM, available from ⟨http://www.omg.org/spec/ARM/1.0/Beta1/⟩.
[19] Inge, J.R.: The safety case, its development and use un the United Kingfom, *Proc. ISSC25*, OMG (2007). available from ⟨http://www.omg.org/spec/SAEM/1.0/Beta1/⟩.
[20] Tim, K. and Rob, W.: The Goal Structuring Notation – A Safety Argument Notation, *Proc. Dependable Systems and Networks 2004 Workshop on Assurance Cases* (2004).
[21] Stephen, E.T.: The Uses of Argument, Cambridge University Press (1958).
[22] The Adelard Safety Case Development (ASCAD), Safety Case Structuring: Claims, Arguments and Evdence, available from ⟨http://www.adelard.com/services/SafetyCaseStructuring/index.html⟩.
[23] DEOS プロジェクト，入手先 ⟨http://www.crest-os.jst.go.jp⟩.
[24] 松野 裕，山本修一郎：実践 D-Case—ディペンダビリティケースを活用しよう！，アセットマネジメント (2014).
[25] 小林茂憲，山本修一郎：アシュアランスケースを用いたサービス提供判断方法の提案，電子情報通信学会，信学技報，Vol.111, No.489, KBSE2011-70, pp.7–12 (2012).
[26] Alexander, R. et al.: Security Assurance Cases: Motivation and the State of the Art, *High Integrity Systems Engineering Department of Computer Science*, University of York (2011).
[27] Goodenough, J. et al.: Arguing Security - Creating Security Assurance Cases (2007), available from ⟨https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/assurance/643-BSI.html⟩.
[28] Lipson, H. and Weinstock, C.: Evidence of Assurance: Laying the Foundation for a Credible Security Case (2008), available from ⟨https://buildsecurityin.us-cert.gov/bsi/articles/knowledge/assurance/973-BSI.html⟩.
[29] Ankrum, T.S. and Kromholz, A.H.: Structured Assurance Cases: Three Common Standards, *Proc. 9th IEEE International Symposium on High-Assurance Systems Engineering HASE'05* (2005).
[30] Bloomfield, R. et al.: Assurance Case Workshop (2005).
[31] Bloomfield, R. et al.: International Working Group on Assurance Cases (for Security), *IEEE SECURITY & PRIVACY* (2006).
[32] Bloomfield, R. et al.: Assurance Cases for Security: The Metrics Challenge, *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks DSN'07* (2007).
[33] Vivas, J.L. et al.: A Methodology for Security Assurance Driven Development, *Requirements Engineering*, Vol.16, No.1, pp.55–73 (2011).
[34] 山本修一郎：主張と証拠，アセットマネジメント (2014).
[35] Yamamoto, S., Kaneko, T. and Tanaka, H.: A Proposal on Security Case based on Common Criteria, Asia ARES2013 (2013).
[36] Kaneko, T., Yamamoto, S. and Tanaka, H.: Proposal on Countermeasure Decision Method Using Assurance Case And Common Criteria, *Promac2012* (2012).
[37] 金子朋子，山本修一郎，田中英彦：セキュリティ保証ケースを用いた対策立案方法の提案，SCIS2013 (2013).
[38] 吉岡信和，Nuseibeh, B.：セキュリティ要求工学の概要と展望，情報処理，Vol.50, No.3 (2009).
[39] IPA（独立行政法人情報処理推進機構）：A 社個人情報処理システムアプリケーションセキュリティターゲット，入手先 ⟨https://www.ipa.go.jp/security/jisec/index.html⟩.
[40] 経済産業省情報セキュリティ対策ポータル，入手先 ⟨http://www.meti.go.jp/policy/netsecurity/cc.html⟩.
[41] IPA（独立行政法人情報処理推進機構）：「IT セキュリティ評価及び認証制度に関する説明会」資料，入手先 ⟨http://www.ipa.go.jp/security/jisec/seminar/cc_semi_20140610.html⟩.
[42] Protection Profiles of Common Criteria, available from ⟨http://www.commoncriteriaportal.org/pps/⟩.
[43] The 2013 International Common Criteria Conference (ICCC) website, available from ⟨http://www.commoncriteriaportal.org/iccc/ICCC_arc/⟩.
Jareno, A.D. et al.: Producing Protection Profile for Internet Banking Application, available from ⟨http://www.fbcinc.com/e/ICCC/presentations/T3_D2_3pm_Jarno_Collab_Efforts_in_Malaysia_to_Product_PP.pdf⟩.
[44] 金子浩之：コモンクライテリアにおけるセキュリティ要求の規定の現状と課題，情報処理，Vol.50, No.3 (2009).
[45] セキュリティ評価方法（CEM バージョン 3.1 リリース 4）情報技術セキュリティ評価のための共通方法，入手先 ⟨http://www.ipa.go.jp/security/jisec/cc/⟩.

金子 朋子 （正会員）

1988 年慶應義塾大学文学部卒業．同年（株）NTT 入社．（株）NTT データにてシステム開発業務に従事．1993 年創価大学法学部卒業．2014 年情報セキュリティ大学院大学情報セキュリティ研究科博士後期課程修了．博士（情報学）．現在，（株）NTT データ 品質保証部所属．情報セキュリティ大学院大学客員研究員．文部科学省認定プログラム・サーティフィケート取得者（情報セキュリティスペシャリスト，ソフトウェアスペシャリスト）．

山本 修一郎 （正会員）

1977 年名古屋工業大学情報工学科卒業．1979 年名古屋大学大学院工学研究科情報工学専攻修了．同年日本電信電話公社入社．2002 年（株）NTT データ　技術開発本部副本部長．2007 年同社初代フェロー，システム科学研究所所長．2009 年東京工業大学統合研究院医療情報プロジェクト特任教授．同年名古屋大学情報連携統括本部情報戦略室教授．ソフトウェア工学，要求工学，IC カードプラットフォーム，ユビキタスコンピューティング，知識創造デザインの研究に従事．情報処理学会業績賞，電子情報通信学会業績賞，逓信協会前島賞等受賞．博士（工学）．著書に，『要求定義・要求仕様書の作り方』（ソフト・リサーチ・センター，2006 年）『～ゴール指向による～システム要求管理』（ソフト・リサーチ・センター，2007 年），すりあわせの技術（ダイヤモンド社，2009 年）CMC で変わる組織コミュニケーション（NTT 出版，2010 年）等．電子情報通信学会知能ソフトウェア工学研究会研究専門委員長 2000～2002 年人工知能学会知識流通ネットワーク研究会主査（2007 年～）．電子情報通信学会，日本ソフトウェア科学会，人工知能学会，日本情報経営学会，ACM，IEEE 各会員．

田中 英彦 （フェロー）

1970 年東京大学大学院博士課程修了，工学博士．東京大学工学部教授，同情報理工学系研究科教授・研究科長を経て，2004 年情報セキュリティ大学院大学教授，研究科長．2012 年同大学学長．計算機アーキテクチャ，分散処理，知識処理，デペンダブル情報システム等に興味を持つ．著書に『非ノイマンコンピュータ』，『計算機アーキテクチャ』，『Parallel Inference Engine』等がある．情報処理学会名誉員，人工知能学会名誉員，電子情報通信学会フェロー，IEEE ライフフェロー．