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

野呂惇[†1]　　松浦佐江子[†1]

システム開発では，システム開発の実装や試験のフェーズでセキュリティ上の問題が発見されることによる設計や要求分析のフェーズへの開発の手戻りが発生する問題がある．こうした問題に対し，システム開発の上流工程でセキュリティの分析を行なうことが求められるが，分析者のセキュリティに関する知識量の差，分析の複雑化等の問題から現状では有効な手法は存在していない．
本研究では，このような問題に対し，情報セキュリティの国際評価基準である Common Criteria（CC）に記述されている要素をセキュリティ知識の拠り所とし，開発現場で広く扱われている Unified Modeling Language（UML）を用いた要求分析モデルを利用し，機能要求とセキュリティ要求を分離して分析を行なった後統合することにより，これらの問題に対処したセキュリティ要求分析手法を提案する．
ケーススタディとして，本学の学習支援サイトへの追加機能を用いて本手法のプロセスを示し，有用性についての考察を行なう．

Model Driven Security Requirements Analysis Method based on Common Criteria

ATSUSHI NORO[†1]　SAEKO MATSUURA[†1]

In the system development, there are problems that occur development reworks to the system design phase or the requirements analysis phase by detected problems of security in implementation phase or system test phase. Security analysis method in upper process of system development for these problems is expected. However, there is no effective method because there are problems such as security knowledge of analyst and complication of analysis.
In this research, we propose security requirements analysis method that used elements of Common Criteria (CC) which is a standard of information security for security knowledge, used requirements analysis models based on Unified Modeling Language (UML) which used in system development, dividing and merging function requirements and security requirement.
Moreover, we show the process of our method with a case study that is additional function for the learning management system in our college and discuss application of our method.

1. はじめに

システムの要求分析において,「様々な人が利用できるサービスに対して,利用者毎に使用できる機能を制限したい」というアクセス制御に関わる要求が生じることは多い．このような要求は，機能要求であると同時にセキュリティ要求の側面を持つため,要求分析が複雑になりやすい.また，分析を行なうためにはセキュリティに関する知識が必要となるため，要求分析結果に誤りや漏れが発生する可能性がある．

要求分析時の分析漏れや誤りは後の工程で開発の手戻りを引き起こす可能性があるため，できるだけ少なくすることが望ましいが，大規模かつ複雑化するシステムのセキュリティ要求分析は，以下の観点から困難である．

第 1 に，一般的にシステムの要求分析者はセキュリティに関する深い知識を持たないため，システムのどこに，どれくらい，どのようにセキュリティを実現すべきかを判断することが難しい．第 2 に，機能要求とセキュリティ要求を同時に分析することは要求分析の複雑化につながり，分離して分析を行なうことは機能要求-セキュリティ要求間のトレーサビリティの低下につながるため，分析結果に誤りや漏れが生じやすい．

本研究では，これらの問題を解決し，システム開発の要求分析時にセキュリティ要求を分析する手法を提案する．

第 1 の問題に対し，本研究では情報セキュリティの国際評価基準である Common Criteria（CC）[1]を用いることで分析者のセキュリティに関する知識の補完を行なう．第 2 の問題に対しては,開発現場で広く使用されている Unified Modeling Language（UML）[2]を用いて，機能要求とセキュリティ要求を分離して分析し統合することで，要求分析の複雑化を防ぎつつトレーサビリティの確保を目指す．

本稿では，まず第 2 章で CC の概要，利用する際の問題点とその解決方針について説明する．第 3 章で本手法の説明を行ない，第 4 章でケーススタディによる本手法の適用例を紹介する．第 5 章で手法の有効性の考察を行ない，第 6 章で関連研究との比較を述べる．最後に第 7 章でまとめと今後の方針を述べる．

2. Common Criteria

2.1 利用上の利点

CC とは，情報セキュリティの国際評価基準（ISO/IEC 15408）である．Part 1 から Part3 までの 3 つの文書から構

* †1 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.

成されており，Part 1 には用語の定義等の概要が，Part 2 にはセキュリティ機能の要件（セキュリティ機能コンポーネント）が記述されている．

セキュリティ機能コンポーネントはクラス，ファミリ，コンポーネントの階層構造を持つ．これらは対象となるセキュリティの関心事をカテゴリ毎に分類したものである．CC は一般的な開発者には理解しづらい文書であるが，クラス，ファミリ，コンポーネントのキーワードは一般的な開発者にも十分理解できるレベルで表現される．表 1 にセキュリティ機能コンポーネントのクラス一覧を示す．

表 1　セキュリティ機能コンポーネントのクラス一覧

クラス	クラス名	クラス	クラス名
FAU	セキュリティ監査	FCO	通信
FCS	暗号サポート	FDP	利用者データ保護
FIA	識別と認証	FMT	セキュリティ管理
FPR	プライバシー	FPT	TSF の保護
FRU	資源利用	FTA	TOE アクセス
FTP	高信頼パス／チャネル		

また，セキュリティ機能コンポーネントは依存関係を持っているため，あるコンポーネントをシステムのセキュリティ機能として扱う場合，そのコンポーネントが依存しているコンポーネントもセキュリティ機能とすることで，あるセキュリティの関心事に対して網羅的なセキュリティの分析を行なうことが可能である．

2.2 利用上の問題点

CC は多様な情報システムのセキュリティ要件を定義するため，高い抽象度を持っているので，これを用いて対象システムに対するセキュリティ要件を定義するためには，CC に一定の解釈を与える必要があるという問題点がある．

本稿では，次のようにセキュリティ機能コンポーネントの解釈と，対象システムのモデル要素との対応を定義する．

アクセス制御に関わるセキュリティ機能要件は，複数のセキュリティ機能方針（SFP）を定義し，それを対象システムに適用することで実現される．各 SFP は，「サブジェクト，オブジェクト，資源または情報，及びそれが適用される操作を定義することにより，そのアクセス制御範囲を特定する」という規則の集合である．ここで，サブジェクト等の用語は CC において定義されており，「サブジェクト」が制御される「操作」の主体を，「オブジェクト」が「操作」の対象を意味する．

一方，セキュリティ機能コンポーネントは，例えば図 1 のようにアクセス制御機能のコンポーネント FDP_ACF.1 として定義されており，4つのセキュリティ機能の要素から構成されている．ここで，FDP_ACF1.1は FDP_ACF.1.2から FDP_ACF1.4の3つのルールで定義される SFP をシステムに適用することが書かれている．規則はサブジェクトやオブジェクトに対するセキュリティ要件を定義するためのセキュリティ属性によって定義される．規則は，これらの関係や，オブジェクトのセキュリティ属性に基づいて，そのオブジェクトを対象とする操作を主体であるサブジェクトに許可または拒否することを決定する．

FDP_ACF.1 セキュリティ属性によるアクセス制御
下位階層: なし
依存性: FDP_ACC.1 サブセットアクセス制御
FMT_MSA.3 静的属性初期化

FDP_ACF.1.1 TSF は、以下の[割付: *示された SFP 下において制御されるサブジェクトとオブジェクトのリスト、及び各々に対応する、SFP 関連セキュリティ属性、または SFP 関連セキュリティ属性の名前付けされたグループ*]に基づいて、オブジェクトに対して、[割付: *アクセス制御 SFP*]を実施しなければならない。

FDP_ACF.1.2 TSF は、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の操作が許されるかどうかを決定するために、次の規則を実施しなければならない: [割付: *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*]。

FDP_ACF.1.3 TSF は、次の追加規則、[割付: *セキュリティ属性に基づいてオブジェクトに対するサブジェクトのアクセスを明示的に許可する規則*]に基づいて、オブジェクトに対して、サブジェクトのアクセスを明示的に許可しなければならない。

FDP_ACF.1.4 TSF は、次の追加規則、[割付: *セキュリティ属性に基づいてオブジェクトに対するサブジェクトのアクセスを明示的に拒否する規則*]に基づいて、オブジェクトに対して、サブジェクトのアクセスを明示的に拒否しなければならない。

図 1　コンポーネント FDP_ACF.1

本研究では，セキュリティ機能コンポーネントに記述されている規則から，規則の判定や実行可能な操作とその主体をアクティビティ図のアクション，パーティション，条件分岐を用いて定義する．また，後述の本手法におけるプロトタイプ生成のために，ユーザとシステムのインタラクションとなる表示に関わるフローをアクティビティ図中に定義する．図 2は FDP_ACF.1をアクティビティ図で定義したものである．ここでは操作の対象であるオブジェクトに対して，FDP_ACF.1.2から1.4の規則を満たしている場合に，サブジェクトが操作を実行し，そうでなければ何もしないことを表している．

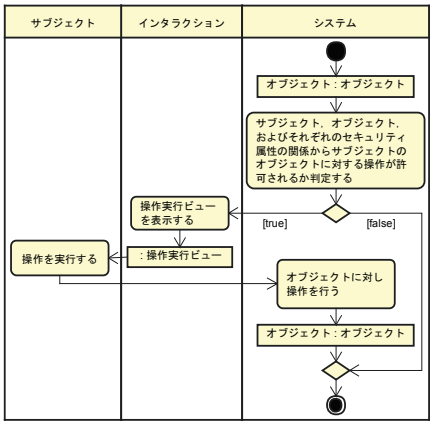

図 2　FDP_ACF.1 を表すアクティビティ図

一方，SFP は，上述の規則の定義要素である，サブジェクト・オブジェクトおよびそれらのセキュリティ属性・操作を，規則を実現する要素として対象システムのモデル要素と対応付けることにより，コンポーネントを構成するセキュリティ機能要件を対象システムの言葉で表現する．

3. 提案手法

本手法の概要を図 3 に示す．

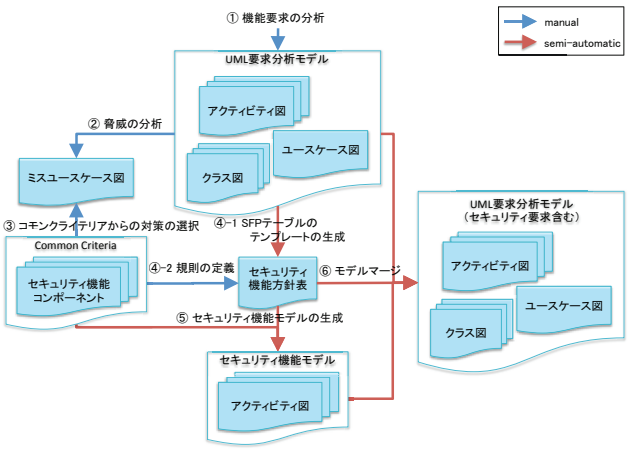

図 3　提案手法の概要

3.1 機能要求の分析

まず機能要求の分析を行なう．本手法では，我々が研究を行なっている UML を用いた要求分析手法である「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」[3]を用いて分析を行なう．これは，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 モデルからツールを用いてプロトタイプ(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．

開発者は分析結果を随時画面イメージとして生成することができるため，分析内容が正しくモデルに反映されているかを確認でき，また顧客との相互確認も可能となる．

この手法による機能の要求分析を行ない，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データおよびエンティティデータ，例外時の振る舞い，データの構造およびその関係と制約を UML モデルにより定義する．

3.2 ミスユースケースによる脅威の分析

機能要求を分析した UML モデルを基に，システムの脅威を分析する．分析はミスユースケース図[4]を用いて行なう．ミスユースケース図とは，システムに対し脅威を与える行為（ミスユースケース）とそれを引き起こすユーザ（ミスユーザ）の関係をユースケース図に加えたものである．

まず，分析対象のミスユーザとなりうるユーザを分析し，3.1 で分析された機能のフローをミスユーザが悪用することによりもたらされるシステムが守るべき情報である資産に対する脅威をミスユースケースとして分析する．

機能要求の分析により，システムで扱う情報はエンティティデータとして定義されているため，分析者は資産をエンティティデータから分析できる．

3.3 コモンクライテリアからの対策の選択

分析した脅威への対抗策となるセキュリティ機能コンポーネントを CC から選択する．CC は「利用者データ保護」や「プライバシー」等のカテゴリで分かれているため，これらのカテゴリと対象の脅威とを対応付け，脅威に対する主要な対抗策となるセキュリティ機能コンポーネントを選択する．また，コンポーネントの依存性を利用することにより，選択した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必要なセキュリティ上の制約を網羅的に分析することができる．

3.4 セキュリティ機能方針の定義

SFP はシステムのサブジェクトがオブジェクトに行なう操作に対する規則の集合である．したがって，システムのサブジェクト，オブジェクト，操作をテーブル形式にまとめ，このテーブルを用いて操作に対する規則を定義することで SFP を定義する．

まず SFP テーブルを UML 要求分析モデルの要素を用いて生成する．それぞれの単語の定義から，サブジェクトは UML モデルのアクタ，オブジェクトは UML モデルのエンティティクラス，操作は UML モデルのアクティビティ図上でエンティティクラスに対する操作を行なっているアクションと対応付け，これらの要素を UML モデルから抽出し，SFP テーブルのテンプレートを自動生成する．

次に，生成された表中の資産となるオブジェクトにセキュリティ属性を付加し，このオブジェクトへの操作に対する SFP の規則を，対策として選択したコンポーネントおよびそれと依存性のあるコンポーネントから定義する．分析者は対象となるコンポーネントのアクティビティ図を参照することで規則の概要を理解することができ，規則を定義するアクションの開始時または終了時の対象オブジェクトのセキュリティ属性の値を規則として定義する．

3.5 セキュリティ機能の生成

選択したコンポーネントと SFP を基に，分析対象システムの脅威に対抗するために必要なシステムのフローをセキュリティ機能として定義する．

まず，アクティビティ図で表したコンポーネントのサブジェクト・オブジェクト・操作・セキュリティ属性を，SFP テーブルで規則を定義したサブジェクト・オブジェクト・操作・セキュリティ属性の項目と置き換える．そして，SFP として定義した規則を条件判定のアクションに対応付け，分析対象システムのセキュリティ機能を自動生成する．

3.6 モデル合成とプロトタイプ生成

定義したセキュリティ機能のモデルを機能の要求分析モデルにマージすることで，セキュリティ機能を含む要求分析モデルを作成する．

SFP テーブルには，セキュリティ上の規則を定義したシステムのサブジェクト・オブジェクト・操作のセットが記述されている．また，これらのセットは既存の要求分析モデルの要素から抽出したものである．したがって，要求分析モデル上のそれらセットが記述されている箇所を SFP テーブルから特定でき，その箇所にセキュリティ機能を埋め込むことで，セキュリティ機能を要求分析モデルにマージ

表 2　Luminous BBS の SFP テーブル

サブジェクト		オブジェクト		操作		規則			
名前	セキュリティ属性	名前	セキュリティ属性	アクティビティ図	アクション	FDP_ACF.1	FMT_MSA.3	FMT_MSA.1	…
学生	役割(学籍番号)	⋮		⋮	⋮				
		話題	公開/非公開	質問を投稿する	話題を生成する		ルールB1		
		投稿者	役割	話題を選択する(学生)	現在の利用者を取得する				
				質問を投稿する	投稿者を取得する				
		添付ファイル	公開/非公開	話題を閲覧する(学生)	添付ファイルをダウンロードする	ルールA			
				質問を投稿する	添付ファイルを投稿する		ルールB2		
教員	役割(教員)	⋮		⋮	⋮				
		話題	公開/非公開	質問に回答する	選択された話題履歴を取得する				
				質問に回答する	回答を追加し話題を更新する				
					公開/非公開を公開に変更する			ルールC1	
					公開/非公開を非公開に変更する			ルールC2	
		投稿者	役割	話題を選択する(教員)	現在の利用者を取得する				
				質問に回答する	投稿者を取得する				
		添付ファイル	公開/非公開	話題を閲覧する(教員)	添付ファイルをダウンロードする				
				質問に回答する	添付ファイルを投稿する		ルールB3		
					公開/非公開を公開に変更する			ルールC3	
					公開/非公開を非公開に変更する			ルールC4	

することが可能である．

また，生成された要求分析モデルから Web-UI プロトタイプを生成することで，マージされたセキュリティ機能を顧客と確認することが可能となる．これにより，セキュリティ要求に関する顧客との相互確認が可能となる．また，セキュリティ要求がマージされたモデルは複雑になりやすいが，プロトタイプを確認することで要求が適切に反映されているかどうか確認できる．

4. ケーススタディ

4.1 Luminous BBS

この章では，ケーススタディとして Luminous BBS への適用事例を用いながら，本手法のプロセスを示す．

Luminous とは本学で使用している授業支援サイトの名称であり，学生・教員間の教材やレポート等のやりとりをサポートする機能を持つ．この Luminous 上で学生・教員間で質問のやりとりを行なうための追加システムが Luminous BBS である．主な機能は学生が質問を投稿でき，教員がそれに回答できること，質問・回答のセット（話題）を閲覧できること，質問・回答にファイルを添付でき，それをダウンロードできることである．

4.2 ミスユースケースによる脅威の分析

Luminous BBS は学生と教員が Luminous にログインすることで使用できるシステムであり，ユーザの成りすまし等の利用者の識別・認証に関わる脅威は稼働中の Luminous により既に取り除かれているため, Luminous BBS を使用できるユーザは学生と教員に限定される．また，教員は管理する立場であるため，学生がミスユーザになり得ると考えられる．Luminous BBS での資産は，質問や回答の内容のセットである話題，履歴の添付ファイルが考えられる．したがって，分析者は定義されたアクティビティ図のフローを悪用することによりこれらの資産が脅かされる学生の行為を考えることで脅威を分析できる. Luminous BBS に対してのミスユースケース分析結果を図 4 に示す．

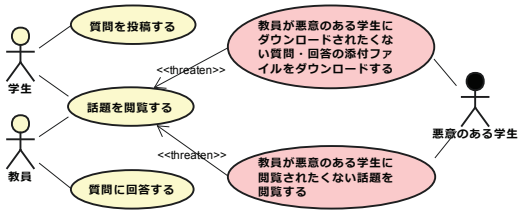

図 4　Luminous BBS のミスユースケース図

4.3 コモンクライテリアからの対策の選択

分析されたミスユースケースはどちらもユーザが利用するデータに対するアクセス制御に関わる脅威であるため，「利用者データ保護（FDP)」クラスの「セキュリティ属性によるアクセス制御（FDP_ACF.1)」のコンポーネントを選択し，それぞれの脅威に対する対策とする．また，FDP_ACF.1 と依存性のあるコンポーネントである「静的属性初期化（FMT_MSA.3)」「セキュリティ属性の管理（FMT_MSA.1)」も，脅威に対する対策として考える．前者はオブジェクト生成時にセキュリティ属性が初期化されなければならないことが，後者はセキュリティ属性に対して適切な役割を持つもののみが必要な操作できなければならないことが記述されている．

4.4 セキュリティ機能方針の定義

まず，モデルから表を生成し，Luminous BBS の資産である話題と添付ファイルのアクセス制御のためのセキュリティ属性として「公開/非公開」を定義する．また，アクセス制御のために個人を特定するための属性として投稿者.役割をセキュリティ属性とする．

次に，規則を定義する．例えば，ミスユースケース「教員が不特定の人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」に対し，コンポーネント FDP_ACF.1 を実現するための規則としては,「添付ファイルが公開であるか，添付ファイルの投稿者である場合のみ添付ファイルをダウンロードできる」ことが必要であるため，添付ファ

イルに対する操作である「添付ファイルをダウンロードする」に対し，FDP_ACF.1 の規則として「アクション開始時，添付ファイル.公開/非公開==公開||投稿者.役割==学生.役割」を定義する．

以上の手順により定義される Luminous BBS の SFP テーブルを表 2 に，規則の一覧を表 3 に示す．表 2 中の規則の空欄箇所は「規則がない」ということを表している．

表 3　SFP 表中の規則一覧

ルールA	アクション開始時，添付ファイル.公開/非公開==公開||投稿者.役割==学生.役割
ルールB1	アクション終了時，話題.公開/非公開==非公開
ルールB2	アクション終了時，添付ファイル.公開/非公開==非公開
ルールB3	アクション終了時，(話題.公開/非公開==公開ならば添付ファイル.公開/非公開==公開||添付ファイル.公開/非公開==非公開)&&(話題.公開/非公開==公開ならば添付ファイル.公開/非公開==非公開)
ルールC1	アクション開始時に話題.公開/非公開==非公開ならば，アクション終了時に話題.公開/非公開==公開
ルールC2	アクション開始時に話題.公開/非公開==公開ならば，アクション終了時に話題.公開/非公開==非公開
ルールC3	アクション開始時に添付ファイル.公開/非公開==非公開ならば，アクション終了時に添付ファイル.公開/非公開==公開
ルールC4	アクション開始時に添付ファイル.公開/非公開==公開ならば，アクション終了時に添付ファイル.公開/非公開==非公開

4.5 セキュリティ機能の生成

4.4 で定義した規則は，FDP_ACF.1 の規則であり，この規則に関わるサブジェクト，オブジェクト，操作，セキュリティ属性はそれぞれ「学生」，「添付ファイル」，「添付ファイルをダウンロードする」，「公開/非公開」である．したがって，これらを FDP_ACF.1 を表すアクティビティ図中のサブジェクト，オブジェクト，操作，セキュリティ属性と置き換える．また，定義した規則を，アクティビティ図中の条件判定アクションに置き換える．以上の手順により，Luminous BBS で添付ファイルのダウンロードのアクセスを制御するセキュリティ機能を生成できる．図 5 に定義したセキュリティ機能モデルを示す．

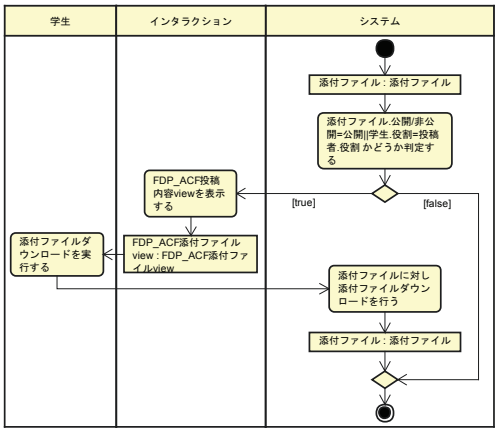

図 5　添付ファイルに対する FDP_ACF.1

4.6 モデル合成とプロトタイプ生成

図 5 は学生のユースケース「話題を閲覧する（学生）」上の「添付ファイルをダウンロードする」操作の実行に必要なセキュリティ機能である．機能モデルとセキュリティ機能モデルの両方に「添付ファイルをダウンロードする」に該当するアクションが存在する．また，[3]による要求分析により，両モデルにはそのアクションを実行するためのアクタパーティション上のアクションと，そのアクションを実行可能にするインタラクションパーティション上の表示アクションが存在する．したがって，これらのアクティビティ図上の構造をキーとして，インタラクションアクションの直前に図 5 に定義されている条件分岐を追加することで，セキュリティ機能モデルを機能モデルのフロー中にマージできる．図 5 をマージしたモデルを図 6 に示す．

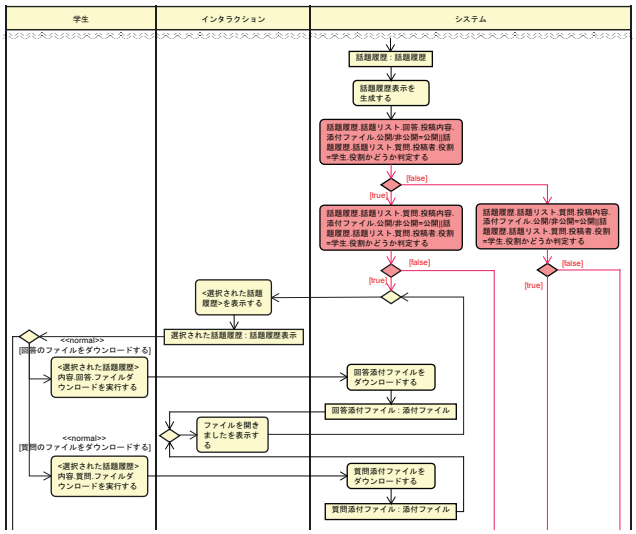

図 6　マージされたモデル例

また，マージされた機能要求モデルから図 7 の Web-UI プロトタイプが生成される．図 7 上は質問の添付ファイルが公開の場合の Web-UI プロトタイプで，図 7 下は質問の添付ファイルが非公開の場合の Web-UI プロトタイプである．開発者または顧客は，これを確認することにより添付ファイルが非公開の場合はファイルダウンロードが実行できないことを確認することができるため，セキュリティ要求が適切に反映されているかを確認することができる．

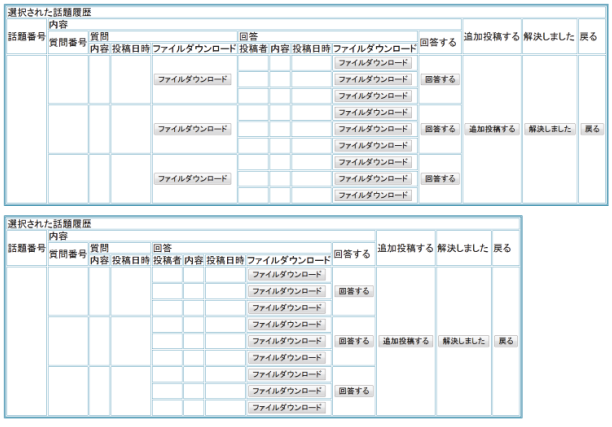

図 7　Luminous BBS の Web-UI プロトタイプ例

5. 考察

第 1 章で述べた第 1 の問題点である分析者のセキュリティ知識に関わる問題に対し，本手法では，CC のセキュリティの知識体系を活用しコンポーネントの依存性を用いることで，分析者のセキュリティ知識を補い網羅的な分析を可能にしている．例えば 4.3 でも述べたとおり，セキュリティ属性によるアクセス制御（FDP_ACF.1）は静的属性初期化（FMR_MSA.3）やセキュリティ属性の管理

（FMT_MSA.1）と依存関係があり，これによりセキュリティ属性への操作の管理や初期値の設定等，アクセス制御に付随するセキュリティの関心事を網羅的に分析できる．

第 2 の問題である要求分析の複雑化に関しては，システムの機能要求とセキュリティ要求を分離し，段階的に分析を行なうことで，複雑化を緩和している．さらに，プロトタイプを確認することで，セキュリティ機能がマージされた複雑なモデルの理解もしやすいと考える．また，トレーサビリティについては，SFP テーブルにより CC のコンポーネントと要求分析モデルを対応付けているため，この表により機能要求とセキュリティ要求のトレーサビリティを確保できる．つまり，どの機能に対しどのセキュリティ機能が関わるかを表から読み取ることができるため，要求の仕様変更等が起きた際も，テーブルから機能要求もしくはセキュリティ要求の変更がどこまで影響を及ぼすのかを理解しやすい．

6. 関連研究

セキュリティ要求分析手法の一つとして，ゴール指向によるセキュリティ要求分析手法がある．

田原らの研究[5]は，ゴール指向分析の一種である KAOS[6]を用いてセキュリティ要求分析を行なうものである．また，Hassan ら[7]は KAOS と形式手法である B method を用いて，ゴール指向分析により導出したセキュリティ要求分析モデルを B method による形式的な設計モデルに変換することを可能とする研究を行なっている．直感的に記述可能なゴール指向分析モデルから形式的な設計仕様モデルに変換することにより，形式的手法の問題点である学習・導入コストの高さを克服し，なおかつセキュリティ要求と設計モデル間のトレーサビリティを確保している．

しかし，これらの手法はゴール指向分析のための業務知識とセキュリティ知識が必要であり，分析結果は分析者の知識背景のみに依存する．また，仕様変更により機能またはセキュリティ要求に変更があった場合，ゴール指向分析ではシステムのどの部分を変更しなければならないかを再分析する必要がある．

我々は，CC のセキュリティ知識を用いた SFP の分析・セキュリティ機能の生成を行なうことで，一般的なシステムの要求分析者のセキュリティ要求分析をサポートする．これにより，分析者の知識背景に依存する分析漏れや誤りを軽減できる．

また，SFP テーブルによりシステムの機能・セキュリティ機能の関係が明確になっているため，仕様変更の際にどの機能またはセキュリティ機能を変更しなければならないかの判断が可能である．

田口らの研究[8]では，CC の認証を受けやすくするために，UML を用いてユースケースや資産となるオブジェクトと CC のセキュリティ機能との関係を表すことができる．これにより，ユースケース図上でシステムのどの機能がどのセキュリティ機能を持たなければならないかを整理できる．しかし，機能のどのタイミングにセキュリティ機能を追加するのかは表現できない．

これに対し，我々は機能のフローにセキュリティ機能のフローを合成することで，具体的に機能のどのタイミングにセキュリティ機能を追加するのかを表現できる．

7. まとめと今後の方針

本稿では，システム開発の要求分析フェーズにおいてセキュリティ要求を獲得し，システムの要求分析モデルに組み込むための手法として，UML 要求分析モデルと CC の要素の対応付けによりセキュリティ機能を定義し，それらの対応関係から要求分析モデルにマージすることを提案した．

本手法では図 3 の④-1，⑤，⑥のフローを自動化できると考えているが，現状では手法全体をサポートできていないため，これに対応するツールを作成し，手法により一般の開発者のセキュリティ知識を補うことができるか評価を行う必要がある．また，本稿では分析された脅威に対して網羅的に対策を定め，それを要求分析モデルに統合することに注力しているが，脅威の分析が網羅的であるか，リスクが十分に検討されているかという課題がある．これらの課題に対し，CC を用いた網羅的な脅威分析や CC で定義されるセキュリティ評価レベルを用いたリスク分析について考察し，これらの課題を改善する必要がある．

参考文献

1) Common Criteria, ”CC/CEM v3.1 Release4”: ISO/IEC 15408, http://www.commoncriteriaportal.org/cc/

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) Sindre, G and Opdahl, A. L. “Eliciting security requirements with misuse cases”, Requirements Engineering Journal, Vol.10, No.2 (2005).

5) 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).

6) 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.

7) Hassan, R.; Bohner, S.; El-Kassas, S.; Eltoweissy, M., "Goal-Oriented, B-Based Formal Derivation of Security Design Specifications from Security Requirements," Availability, Reliability and Security, 2008. ARES 08. Third International Conference on , pp.1443,1450, 2008.

8) K. Taguchi, N. Yoshioka, T. Tobita, H. Kaneko: Aligning Security Requirements and Security Assurance Using the Common Criteria, in Proceedings of IEEE International Conference on Secure Software Integration and Reliability Improvement (SSIRI) '10, pp69-77, 2010.