B-007

ゴール指向を用いたセキュリティ要件の定義手法の提案
Goal Oriented Analysis Method for Security Requirements

府川　真理子† 松浦　佐江子‡
Mariko Fukawa† Saeko Matsuura‡

1. はじめに

要求には利用者がシステムに機能として求める機能要求と明確な定義方法は確立していない非機能要求が存在する．セキュリティは非機能要求として定義される．セキュリティ要件を満たすシステム開発の大半はセキュリティに関わる知識に依存する．そこで，セキュアなシステムを開発するためには，国際標準規格 ISO/IEC 15408 として Common Criteria (CC)[1][2] に定義されたセキュリティ要件を 1 つの規範とすることができる．しかし，CC には膨大な量のセキュリティ要件が定義され，更に 1 つのセキュリティ要件は他の複数のセキュリティ要件との依存関係をもつ為，文書における各項目の依存関係を理解し，適切なセキュリティ要件を選択することが難しい．また，CC に定義されているセキュリティ要件は抽象的な自然言語で記されているためセキュリティを熟知していないシステム開発者では目的となるセキュリティ要件を実現する具体的方法を定義することが困難である．

本研究ではゴール指向により，CC の文書に記載された内容をその非機能要求を達成する基本的な手段（機能要求またはその例示）へと分解過程を読み取り可能とするため，CC を構成するファミリをゴール木として定義する．ゴール指向とはゴール（目的）間の依存関係を明確化する手法であり，ある目的をもつ非機能要求から，その目的を達成するための具体的な手段を系統的に整理することができる．代表的な手法としては i*フレームワーク[3]と NFR(Non-Functional Requirements)フレームワーク[4]がある．ゴール木から非機能要求を機能要求へと分解する過程を読み取ることを可能にすることで，システム開発者がセキュリティ要件を定義する際のガイドラインを作成し，そのゴール木をベースにセキュリティを熟知する開発者のノウハウを整理することを目指す．

本稿の次節以降の構成は下記のとおりである．2 節ではセキュリティ要件と CC の概要，CC の問題点．3 節で NFR フレームワーク，作成するゴール木の解釈，CC からゴール木への分解方法．4 節では CC からゴール木への作成例を利用して分解方法を説明する．ここでは，「識別と認証」の要件である「認証失敗（FIA_AFL(Functional Identification and Authentication_Authentication FaiLures)）」をゴール木で定義する．5 節では「ＷＥＢバンクのパスワード認証」におけるセキュリティ要件が定義可能であるかを検証する．そして，6 節では関連研究について述べ，最後に問題点の考察と今後の課題を述べる．

† 芝浦工業大学大学院　電気電子情報工学専攻
Shibaura Institute of Technology Graduate School of Engineering Electrical Engineering and Computer Science
‡ 芝浦工業大学　システム理工学部
電子情報システム学科
Shibaura Institute of Technology College of Systems Engineering Department of Electronic Information System

2. セキュリティ要件の構成と問題点

2.1 セキュリティ要件と CC

セキュリティ要件とはセキュリティシステム開発時に満たすべき条件や要件間の依存性の定義で CC に定義されている．CC とは評価基準であり，一般法則，セキュリティ機能要件，セキュリティ保証要件の 3 パートから成る．今回使用する要件は CC のパート 2「セキュリティ機能コンポーネント」である．これは開発対象システムのセキュリティ機能要件の基となる標準テンプレートとして，機能コンポーネントのセットをカタログ化しており，11 種類のクラスと，総計 65 個のファミリから成る．例えば，クラス「識別と認証」には，「認証失敗」等 6 個のファミリが定義され，「認証失敗」にはコンポーネント「認証失敗時の取り扱い」が定義されている．コンポーネントには自己完結する要素としてエレメントが定義され，クラス・ファミリ・コンポーネント・エレメントと階層構造となる．各階層は自然言語にて段落分けされ記される．

セキュリティ機能要件にはこれらの定義（以後，本文と呼ぶ）の他に，本文の具体的な要件を例示する等セキュリティ要件定義者が本文を理解するための適用上の注釈である附属書がある．このとき，附属書も本文同様の構成となる．また，CC には定義の他に管理・監査の側面からの要件や依存関係も定義されている．このように，CC は意味的構造を持ち階層的に定義されている

2.2 CC の問題点

CC の問題点として次の①～④が挙げられる．

①要件はクラス間で依存する

②附属書は本文との対応関係を読み解く必要がある

③セキュリティを熟知していないシステム開発者では下記のようにエレメントに[割り付け]や[選択]と記された抽象的な値を，対象システムに合う具体的な値にして要件を完成させることが困難となる．割付とはコンポーネントまたは要件内の識別されたパラメタの特定を表す．パラメタとは制限のない変数，または変数を特定の範囲の値に狭める規則を表す．また，選択はコンポーネント内のリストから 1 つまたは複数の項目の特定を表す

④あるシステムのセキュリティ要件定義者以外がセキュリティ仕様書を見た際，要件を採用/不採用とした理由が不明瞭となる．

CC に記されているエレメント（未完成）

TSF は[割付:認証事象のリスト]に関して，[割付:回数]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

TSF：評価対象のセキュリティ機能

完成する要件

TSF は**使用者のパスワード**に関して，**4** 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

CC は意味的構造を持ち階層的に定義されているため，ゴール指向分析の手法として，非機能要求を目的から手段へ階層的に分析する NFR フレームワークを採用する．

3. ゴール指向の解釈と分解方法

3.1 NFR フレームワーク

NFR フレームワークにおける非機能要求を表現する基本単位として，非機能要求の満足化を表現する NFR ソフトゴール，満足化を助ける技術を表現する操作ソフトゴール，意思決定の根拠を表現する理由ソフトゴールがある．そして，ソフトゴール間の関係として洗練関係と貢献関係がある．洗練関係とは階層の縦と横の関係を表す．貢献関係とは上位ソフトゴールの満足化に下位ソフトゴールが貢献する関係を表す．貢献関係の種類として AND 関係，OR 関係，そして Satisficing（満足化）関係がある．満足化関係とは上位ソフトゴール満足化のために下位ソフトゴールが 1 つだけ必要，或いは下位ソフトゴールが必ずしも必要でない関係を表す．図示するに当たり UML/MDA プラットフォームの StarUML[5]を利用する．

一般的に NFR フレームワークを用いた要求分析は 1～4 の手順で行う．今回は，手順 2 の作成方法を提案する

1 満足化すべき非機能要求を顧客に文章等で表現する．
2 最初に分かる非機能要求を列挙し，
　複数のゴール木を作成する．
3 複数のゴール木間の相互干渉を明確にする．
4 ゴール木の終端を評価し満足化を確認する．

3.2 作成するゴール木の方針

3.2.1 全体の方針

今回提案するゴール木は人間が横文字を読む際，同様に左から右方向へと順に要件を定義するため，要件の依存性が大きいソフトゴールほど左に定義する．

CC はシステムの振る舞いについて評価者の観点からの記載であるため，利用されるデータや実行する操作を整理してシステムを作成する開発者には分かりづらい構成となる．そこで，開発者に馴染みのあるプログラムの形としてオブジェクト指向のクラスの概念をファミリに適応する．クラスには利用されるデータとしての属性と，振る舞いとしての操作がある．

3.2.2 階層の判定基準

ゴール木は上位階層からクラス，ファミリ，コンポーネント，エレメントと CC に定義された構造に従う．

全体の解釈より，ファミリの記述を属性と操作の 2 つに分類し，これをファミリの下位階層に定義する．このとき，操作が属性に依存する場合を考慮し，属性を左側に操作を右側に定義する．更に属性は属性の種類を持ち，同様に操作も操作の種類である振る舞いすなわち，コンポーネントを持つ．これを対応する下位階層に定義する．そして，属性の種類には属性の値と範囲があると考え，属性の種類の下位階層としてこの 2 つを定義する．同様に振る舞いにはそれを行う手段があると考え，振る舞いの下位階層に定義する．このとき，対応するソフトゴールが存在する場合は必ず定義するため，全ての貢献関係は AND 関係となる．階層の定義により，ソフトゴールの階層の妥当性を判断する基準ができる．

3.3 CC からゴール木への分解方法

3.3.1 全体の分解方法

ファミリ全体のゴール木の作成手順を A～F に示す．このとき，各階層は自然言語にて段落分けされ記述される．

A 本文のファミリについてゴール木を作成
B 附属書のファミリについてゴール木を作成
C 本文のコンポーネントについてゴール木を作成
D 附属書のコンポーネントについてゴール木を作成
E 本文のエレメントについてゴール木を作成
F 附属書のエレメントについてゴール木を作成

B～F で作成するゴール木を，A を元とする既に作成されたゴール木に追加することで，最終的に本文・附属書両者の要件を満たした 1 つのゴール木として定義する．これにより，問題②は緩和される．

3.3.2 段落毎の分解方法

各階層は自然言語にて段落分けされ記される．CC の段落分けされた特徴を利用し，段落を 1 つの構造として考える．そして段落にはいくつかの文があるため，文ごとにゴール木を作成する．文をゴール木へ分解する際には階層の判定基準と分解ルールを利用する．このねらいは，誰でも一意にゴール木を作成可能することである．段落毎のゴール木の作成手順を Ⅰ～Ⅲに示す．

Ⅰ 文からソフトゴール単体を定義する
Ⅱ ソフトゴール間の貢献関係を決定する
Ⅲ ソフトゴールの洗練関係を決定する．

3.3.3 分解のルール

CC は抽象的な記述のため，ソフトゴールの種類は例示と理由以外は全て NFR ソフトゴールにて表現する．以下に今回利用したルールを示す．表 1 表 2 は全ての自然言語に対して適応される．表 3 はエレメントにのみ適応される．

	対象	貢献関係	親	子	個体
1	Aは・Aについての要件が含まれる ～を定義する（包含）	不明	A(主語)	述語	×
2	A及びB，A・Bを要求する	AND	不明	A,B	×
3	AまでBする，Bの後Aする（手段）	Satisficing①	A	B	×
4	AあるいはB，AかB	OR	不明	A,B	×
5	（文A）また（文B）（つなぎの接続詞）	不明	文A	文B	×
6	例えば，Aなど（例示）	Satisficing	具体例	A(例)	(例)(操作ソフトゴール)
7	親ゴールGに対してG=A∧（B∨C）	AND	G	E(B∨C)	E(B∨C)
8	親ゴールGに対してG=A∨B∧C	OR	G	E(B∧C)	E(B∧C)
9	Aの場合Bする（場合分け）	不明	A	B	×
10	AするためBする（目的手段）	不明	A	B	×
11	A・B・Cが同列に定義されている	不明	不明	A,B,C	×
12	AのためBする（理由）	不明	不明	B	A(理由ソフトゴール)
13	文の動詞が同じ	×	どちらかが親あるいは子となる		×

表 1 文をゴール木に変換するルール

	対象	貢献関係	親	子	個体
1	既に定義されたソフトゴール名が同意	×	×	×	差分を採用して上書きする
2	PP/ST作成者以外がある動作を許可される	×	×	×	「許可された役割名」(許可された動
3	（条件A）になる（条件）	×	×	×	検出する
4	パラメタとして定義された要素がある	×	×	×	<パラメタとして明当てはまる部分>

表 2 文をソフトゴール単体に変換するルール

	対象	貢献関係	親	子	個体
1	エレメントにおける割り付け	Satisficing	割り付ける対象	割り付けるもの	×
2	エレメントにおける選択	OR	選択対象が示すもの	選択対象1，選択対象2	×

表 3 エレメントに関するルール

ルールについて説明する．まず，「不明」とは文により異なる意味を持つため，明示的に表すことが出来ない状態を表す．また「×」とはルールを適応するのみでは定義できない状態を表す．「対象」列はルールの適応対象となる自然言語あるいは状態を表す．「貢献関係」列はルールを適応した際の貢献関係を表す．「親」「子」列はルールを適応した際の親ソフトゴールと子ソフトゴールを表す．

「個体」列はルールを適応した際，新たにソフトゴールを作成，或いはソフトゴール名を変更する状態を表す．

表 2 において，パラメタを含むソフトゴールを定義する際，パラメタが評価対象機能に与える影響が大きいためパラメタの値をリスク管理などで熟慮して定義する必要がある．また，手順 A は，B～F で利用する表 1~表 3 の他に特殊ルールが存在する．次章にソフトゴールへ分解する方法を CC に定義された認証失敗のゴール木の作成例と共に示す．認証失敗とは認証が失敗する際に行うべき動作とそれの終了条件について定義する．

4. CC からゴール木への作成例

4.1 手順 A：ファミリのゴール木作成

CC のファミリの記載の一部を以下に転載する．このとき冒頭番号は段落を，《 》はソフトゴール名を表す．

> 242 このファミリには、不成功の認証試行の回数に関する値、及び認証の試行が失敗した場合の **TSF** アクションの定義についての要件が含まれる。

まず，手順 I よりソフトゴール単体を定義する．特殊ルールとして，まずファミリとして《FIA_AFL 認証失敗のファミリの振る舞いを定義する》というソフトゴールを定義する．次に全体の解釈より，属性と操作になりうる部分をそれぞれ《〈不成功認証試行の回数〉に関する値を定義する》《認証試行が失敗した場合の TSF アクションを定義する》と定義する．次に手順Ⅱより貢献関係を決定する．全体の解釈より，ファミリは属性と操作を持つため，作成した属性と操作に対応する 2 つのソフトゴールをファミリの下位ソフトゴールとして貢献関係 AND で定義する．最後に手順Ⅲより洗練関係を決定する．階層の判定基準より属性に当たるゴールを左に，操作に当たるゴールを右に定義する．以上から図 1 のゴール木が作成される．

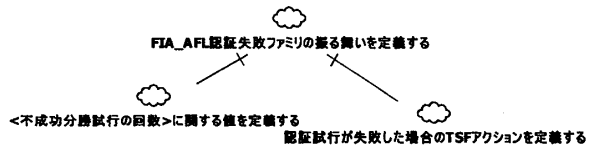

図 1　作成したファミリのゴール木

4.2 手順 B～F：ゴール木作成と追加

4.2.1 段落毎のゴール木の作成

CC のエレメントの記載の一部を以下に転載する．このとき冒頭番号は段落を，《 》はソフトゴール名を表す．

> **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

手順 I よりソフトゴール単体を定義する．まず単語ごとにソフトゴールに分解する．次に，表 2 に適応する状態がある際にはそれを考慮する．最後に文全体の意味についてのソフトゴールを定義する．次に手順Ⅱより，表 1，或いは表 3 に適応するルールを利用してゴール木の貢献関係を決定する．最後に手順Ⅲより，階層の判定基準から洗練関係を決定する．図 2 に作成したゴール木の内の 1 つを示す．

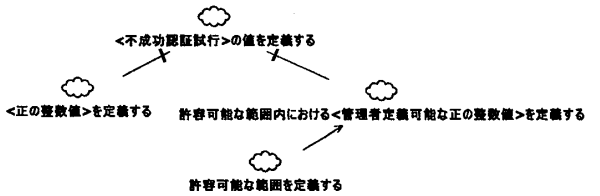

図 2 作成したエレメントのゴール木

4.2.2 作成したゴール木の追加

次に，文から作成したゴール木を既に作成された元のゴール木に追加する．まず，追加する場所の選定を行い，その後追加方法の決定をする．追加する場所の選定とは，作成したゴール木について階層の判定基準を用いて分類する．追加方法の決定では文から作成したものがソフトゴール単体の場合と，ゴール木の場合で振る舞いが異なる．前者はソフトゴール名自体を補足変更し，後者は既に作成されたゴール木について，横・上の階層（目的と手段の関係）を意識して追加する．

図 2 に示したゴール木のルートを見ると「～値を定義する」と記されている．階層の判定基準より，これは属性の種類である不成功認証試行回数の値と関連する．図 3 の赤枠出囲まれていない既に作成された元のゴール木の同階層には，既に同意のソフトゴール名が定義されているため，この下位ソフトゴールに図 2 の作成したゴール木を追加する．このとき，図 3 の上段に示した既に作成された元のゴール木のソフトゴール名と図 2 の作成したゴール木のソフトゴール名には差異がないため，図 3 の下段のように結合した．このとき，図 2 で作成したゴール木を点線枠で示す．

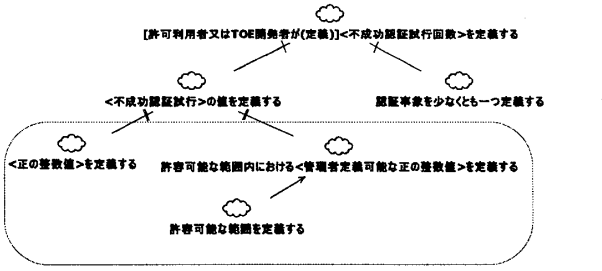

図 3 元のゴール木と追加後のゴール木

4.3 ゴール木の利用

4.3.1 作成したゴール木の読み方

抽象的なゴール（目的）を達成するためには，それをルートとする木を辿り，下位にある具体的な手段を選択・決定する．このとき選択・決定する要求をゴール木の階層毎に比較検討する．

4.3.2 権限を持つ役割

要件定義者を CC では PP/ST 作成者と呼ぶ．また，CC ではその他に要件を定義/管理する権限をもつ役割が定義されている．あるソフトゴールに対して，権限を持つ役割の表記は表 2 に示す．開発者が権限を持つ場合はシステムに組込むため新たにサービスを定義する必要はないが，許可利用者（例えばシステム管理者）が権限を持つ場合は利用者がシステムの外から操作可能とする必要があるため「変更する」等のサービスを新たに定義する．

5. 作成したゴール木の有効性の検証

5.1 前提条件

具体例として実在する銀行の WEB バンクシステムを挙げる．このとき，下記の前提条件があるとする．《 》はソフトゴール名，【 】は前提条件，『 』は結果として求められるセキュリティ要求を表す．

WEB バンクシステムは個人のユーザが個人の PC により利用するため同じものを複数の人と共有しない．従って【一人の利用者がサービスを停止された場合，他の利用者のサービスは停止しない（以後，前提 1）】また，一般的に WEB バンクとは金銭扱うため【セキュリティの十分な確保が必要（以後，前提 2）】となる．前提 2 は認証方式の決定，認証試行回数の決定が挙げられるが，今回は,少なくとも必要な要件として認証失敗時の取り扱いを検証する．そして，実在する要件の内の 1 つとして『ID がロックされると利用者はサービスを受けることができない（以後，結果条件）』がある．今回はこの結果条件を作成したゴール木に前提条件を付加させることで定義可能か検証する．

5.2 検証

結果条件に関する箇所から始める．全体の解釈よりゴール木の右側は操作に関する要件を表す．そこで、結果条件は操作に関るため右のソフトゴールを辿る．すると《セション確立プロセスを無効にする》に辿り着く．セションとは利用者との対話（窓口）を表す．これはサービス拒否と同義である．従って検証するにあたりこれをルートとする部分木を利用する．図 4 に利用するゴール木を示す．

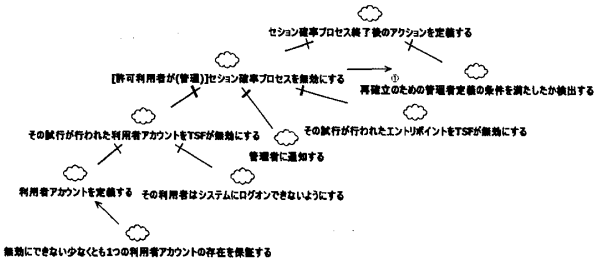

図 4 検証で利用するゴール木

《セション確立プロセスを無効にする》の下位ソフトゴールには《その試行が行われた利用者アカウントを TSF が無効にする（以後，「利用者アカウント禁止」）》《管理者に通知する》《その試行が行われたエントリポイントを TSF が無効にする》がある．前提 1 より複数の人が共有するエントリポイントは無効にしてはならないため，要件を定義する候補が 2 つに絞られる．また，2 つのソフトゴールの貢献関係は両者 OR 関係なためどちらも選択可能である．そこで，左側に定義された「利用者アカウント禁止」，採用し，その下位ソフトゴールである《利用者アカウントを定義する》《利用者はシステムにログオンできないようにする（以後，「ログオン禁止」）》について先程同様吟味する．両者の貢献関係は AND 関係のため必ず選択しなくてはならない．そこで，左側に定義された《利用者アカウントを定義する》を選択し，その下位ソフトゴールである《無効にできない少なくとも 1 つの利用者アカウントの存在を保証する（以後，「アカウント存在」）》に辿り着く．結果，「アカウント存在」という要件が定義される．

同様に「ログオン禁止」「管理者に通知する」の 2 つの要件も残りの要件から定義される．以上からゴール木と前提条件から 3 つの要件を定義し，目的であった結果条件は「利用者アカウント禁止」と「ログオン禁止」を合わせた要件と同義と判断した．更に，ゴール木の利用により「アカウント存在」という要件も定義できた．

6. 関連研究

セキュリティ要件を熟知してないシステム開発者に対してセキュリティ要件を定義可能とする類似の目的を持つ関連研究[6]がある．関連研究では CC のセキュリティを熟知しているシステム開発者が前提として要件を採用する．そして，セキュリティを熟知していないシステム開発者は採用されたセキュリティ要件から対応するセキュリティ要件を定義する．しかし，採用する理由や根拠，採用箇所が関連研究では不明瞭である．また，エレメント以外はセキュリティ要件として扱わないため役割，尺度，条件を考慮することができない．そこで，提案手法では CC に記された本文と附属書を 1 つのゴール木で表すことにより，セキュリティを熟知していないシステム開発者でも CC に記された要件を網羅的に表現できる可能性を示した．

7. まとめ

今回我々は認証失敗の例を用いて，CC からゴール木を一意に作成する手法を提案した．更に，WEB バンクの例を用いて作成したゴール木から要件の網羅的な確認や，要件の根拠を捉える可能性を示した．これにより，セキュリティを熟知していないシステム開発者に対してセキュリティ要件を考慮することが可能となると提案した．

今後の課題として，他のゴール指向との比較による NFR フレームワークの妥当性の検証，CC を熟知していない開発者によるゴール木作成実験によるルールの検証，別の具体例を用いたゴール木の検証，或いは，CC を熟知していない開発者の実験によるゴール木の有効性の検証がある．

また，提案したルールでは CC に記された用語の説明や，自然言語の使い分けがゴール木への変換により消えてしまうため，別途表現方法の検討や，ソフトゴールを定義した際その種類を明確に決定することが今回の提案では出来ないと判断されるため，それについても検討する必要がある．更に，作成したゴール木が膨大となることが懸念されるため，ツールを作成して対処することを検討する．

参考文献

[1] 情報処理推進機構,セキュリティ評価の為のコモンクライテリア　パート 1:概説と一般モデル 改訂第 1 版 http://www.ipa.go.jp/security/jisec/cc/documents/CCPART1V3.1R1-J1.2.pdf
[2]情報処理推進機構,セキュリティ評価の為のコモンクライテリア　パート 2:セキュリティ機能コンポーネント http://www.ipa.go.jp/security/jisec/cc/documents/CCPART2V3.1R2-J2.0.pdf
[3] i*homepage.http://www.cs.toronto.edu/km/istar
[4]L.chung, B.Nixon,E.yu, and J.Mylopoulos, Non-Functional Requirements in Software Engineering. Academic Publishers,1999.
[5]Non-Functional Requirements Modeling Tool http://www.utdallas.edu/~supakkul/tools/softgoal-profile/softgoal-profile.html
[6]大黒博昭　市原尚久　吉村俊哉　明関恵美，メタモデルを用いたセキュアシステム開発支援ツールの開発，電子情報処理学会研究報告 Vol.2006，No.35pp 167-174(2006)