Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

# 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

府川真理子[†]　　松浦佐江子[††]

システムのセキュリティ要件を定義する際，セキュリティ確保に必要な要件の評価基準を定めている国際標準規格 ISO/IEC 15408(Common Criteria：CC)を満たすことは重要である．しかし抽象的な自然言語により記載された文書である CC を基にした要件の管理は困難であるという問題がある．本稿では要件の管理の容易化を図るため，ゴール指向分析により CC のファミリに記載された自然言語をゴールへ分解する規則を定義し，CC 文書の記載内容をゴール木として定義する．そしてゴール木から CC を満たした要件が定義可能か実例を用いて検証する．

# Goal Oriented Analysis Method for Security Requirements

Mariko Fukawa[†]　Saeko Matsuura[††]

It is a good plan to specify system requirements for security functions based on ISO/SEC 15408 (Common Criteria), because it gives us an appropriate criterion to develop a secure system. However, it is difficult to manage the requirements based on CC that was written in abstract natural language. This paper proposes a goal tree that expresses all the contents of CC by a goal-oriented analysis method. Our aim is that the goal tree enables us to reuse the know-how to specify appropriate security requirements. Moreover, we show that an actual example security requirement can be specified by the goal tree.

## 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] 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．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[2]と NFR(Non–Functional Requirements)フレームワーク[3]がある．

本研究ではゴール指向分析により，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）へと分解する過程として表現できるように，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した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以下のとおりである．2 節ではセキュリティ要件の概要，CC の構成，ゴール木の構成要素，セキュリティ要件のゴール木への分解方法といったゴール分析について述べる．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．3 節では 2 節で作成したゴール木により，「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抽出可能であるかを検証する．最後に問題点の考察と今後の課題を述べる．

## 2. ゴール分析

### 2.1 セキュリティ要件の構成と問題点

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 とは評価基準として定義されたもので，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．今回使用した要件は 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 個のファミリで構成されている．例えば，クラス「識別と認証」には，「認証失敗」等 6 個のフ

† 芝浦工業大学院　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

Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

ァミリが定義され,「認証失敗」ファミリ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したエレメントが定義されており，クラス・ファミリ・コンポーネント・エレメントという階層構造になっている．セキュリティ機能要件はこれらの定義（以後，定義の要件と呼ぶ）の他に，その適用上の注釈である付属書（以後，これを付属書の要件と呼ぶ）がある．さらに，定義の要件には，定義の他に管理・監査等の様々な側面や依存関係も定義されている．つまり，一見すると意味的構造を持ち階層的に定義されているようにみえる．

しかし要件はクラス間で依存しているという問題や，付属書は具体的な要件を例示する等，セキュリティ要件定義者が定義の要件を理解しやすくするための注釈として有益であるが，定義の要件に対応付けて，その関係を読み解かなければならないという問題がある．エレメントに記載された要件は以下のように[割り付け]や[選択]と記されているところをシステムに合う具体的な値に書き換えるためそのセキュリティ要件を定義したシステム開発者以外がそのシステムのセキュリティの仕様書を見た場合，どのような理由からこの要件を採用/不採用としたのかが不明瞭となるという問題もある．以上の問題から今回分析する際に，意味的構造を持ち階層的に定義されているという CC の特徴から，ゴール指向分析の手法として，非機能要求を階層的に分析することが可能な NFR フレームワークを採用した．

- 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
- 完成した要件
  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

今回は CC に定義された認証失敗（FIA_AFL）のゴール木を作成した．認証失敗とは CC の識別と認証（FIA）クラスに定義されたファミリの一つである．この要件は認証が失敗した際に行うべき動作とそれの終了条件について定義している．例えばパスワード認証機能において，パスワード入力をある一定回数間違えて入力した際，システムは管理者定義の条件になるまでパスワードの入力を受け付けない．という要件はここに含まれる．

### 2.2 NFR フレームワークの構成要素と貢献関係

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．ソフトゴール間の関係を表すものとして洗練関係と貢献関係を用いる．洗練関係とは上位ソフトゴールを下位ソフトゴールへ展開することを表す．貢献関係とは上位ゴールの満足化に下位ゴールが貢献することを表す．貢献関係の種類として上位ソフトゴールの満足化のためにすべての２つ以上の下位ソフトゴールが必要の場合は AND 関係，上位ゴール満足化のために 2 つ以上の選択肢の内少なくとも 1 つの下位ソフトゴールが必要な場合は OR 関係，上位ゴール満足化のために下位ゴールが 1 つだけ必要あるいは必要ない場合は Satisficing（満足化関係）と呼ぶ．作成するに当たり UML/MDA プラットフォームの StarUML[4]を利用した．

### 2.3 セキュリティ要件のゴール木への分解

一般的に NFR フレームワークを用いた要求分析は下記のように行われる．

① 満足化するべき非機能要求を顧客に文章などで表現する．
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の明確化する．
④ ゴール木の終端を評価・抽出し満足化を確認する．

今回は CC という仕様書を用いるため，①の作業を行う必要はない．CC の文書の構造に従い，手順②により「識別と認証」の「認証失敗」の非機能要求のみをゴール木として表したが,「識別と認証」に定義されている他の要求についてもゴール木を作成する必要がある．手順③に関しては構造化を意図した CC を用いたので既に明確化されている．しかし，依存関係等の把握は困難なために各要件間の相互干渉を明確化する必要はある．手順④は整理されていない終端の項目を整理するための作業であり，今回は CC という整理された情報をカタログ化することが目的であることから，この作業は行わない．

文献[5]では電子タグ保護ガイドラインから分解方法のルールを用いてゴール木の作成を行っている．CC もガイドラインと同じく規約文書であることから，各記述文の分解にはこのルールを適用することとした．

更に上記の文献はガイドラインを分解するために定義されたものなので，今回ゴール木を作成するにあたり貢献関係や名前の付け方を新たに CC に基づく定義を考えた．以下に新たに追加したルールを説明する．なお《　》はソフトゴール名を表す．

- 条件文に関わる定義

条件となる文章のときはそれを満たしたことなどを必ず検出する必要があるため．～検出するという要件を定義する．例えば“管理者定義の条件になるまで”と記されている場合は《再確立のための管理者定義の条件を満たしているか検出する》というソフトゴールを作成する．しかし“利用者アカウントが無効にされた場合はその利用者はシステムにログオンできない”などの検出する対象がすでにソフトゴールとして定義されている場合は,《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》

Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

というソフトゴールのサブゴールとして記されているので《利用者はシステムにログオンできないようにする》と定義する

● ファミリの振る舞いの解釈

ファミリの振る舞いの要件を 1 つのクラスとして考え，属性と操作に分けた．このときコンポーネントには取り扱いなどが記されているため操作として扱う．そして属性は属性の名前や属性役割が定義され，操作にはその操作を実現するための手順が定義されているとする．このとき元からある AND 関係，OR 関係，満足化関係の他に操作の手順としての流れを表現するものとして Satisficing①という関係を定義した． 図 1 に例を示す．


図 1 Satisficing①の例

● 条件による選択

CC の中でなにかの条件により選択をすることが明示的に定義されている場合は，理由ソフトゴールを用いて表記した．例えば図 2 の場合，《通常状態に戻すアクションを定義》する際に，もし《エントリポイントを無効にしている》場合は《エントリポイントを有効にする》必要があることを表しているが，エントリポイントを無効にしていない場合についての要件は表していないとする．このとき点線で表現された雲を理由ソフトゴール，太枠で囲まれた雲を操作ソフトゴール，それ以外を NFR ソフトゴールとして表現する．

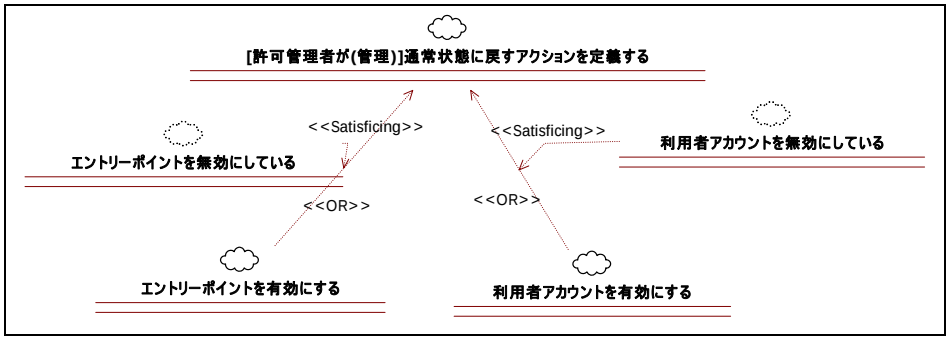


図 2 条件による選択の例

● 異なる種類の貢献関係の表記

（A∨B）∧C という条件がある時には A∨B のゴール木の上に D という上位ゴールを置き D∧C として考える．図 3 に例を示す．

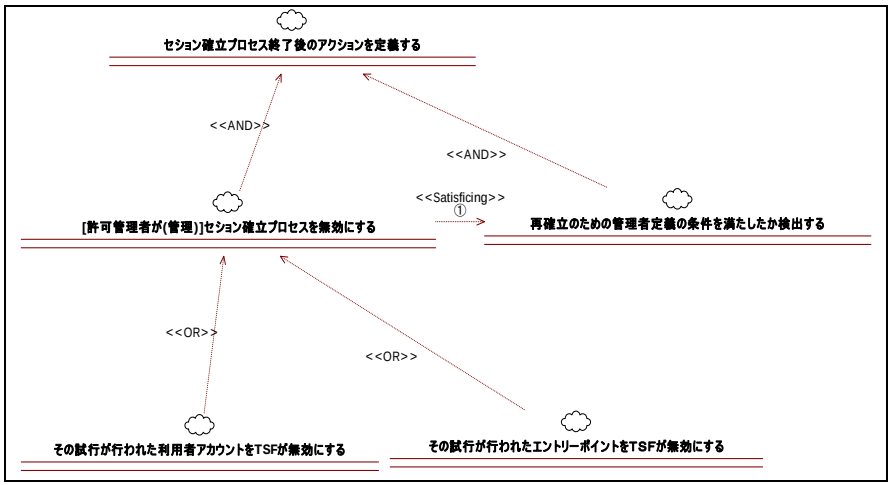


図 3 （A∨B）∧C＝D∧C

● ソフトゴール名の定義

ソフトゴール名は基本的に CC の文中の言葉を用いるが①例として挙げられている②権限を持つ役割が決められている③パラメタが関わる場合は以下のように表現する．

① 例として挙げられているについて

CC 本文中に例として挙げられたものを定義する際にソフトゴール名の前に「(例)」と記述し，操作ソフトゴールで定義する．そして例の対象となるソフトゴールの下位ゴールとして定義する 図 4 に例を示す．

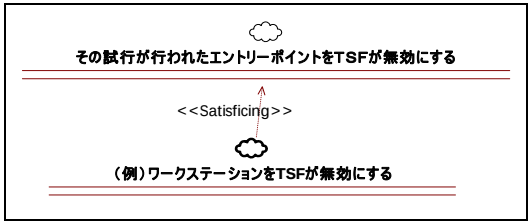


図 4 例の表記

Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

② 権限を持つ役割が決められている場合について

“CC の本パートでは，利用者が操作を行うために必要な権利及び/または特権を有することを示すために，「許可」という用語を使用する．したがって，「許可利用者」という用語は，利用者が SFR によって定義される特定の操作または操作のセットを実行できることを示す．”（CC27 から転載）のように CC には権限というものが定義されている．CC 本文中に「～が」と明示的に記述されている際には，その主体となる役を意識する必要がある．例えば，CC の本文中の“PP/ST(Protection Profile /Security Target)作成者は，不成功認証試行回数を定義することができ，あるいはその回数の定義を TOE(Target Of Evaluation)開発者または許可利用者に任せることを選択できる．”（CC938 から転載）という記述は図 5 のように定義される．ソフトゴール名の前の[ ]中にその権限を持つ役（この場合だと許可利用者，TOE 開発者）を明記する．そして役の後の，() 中にその役が対象ソフトゴールに対して許可された操作を記述する．このときセキュリティ要件を定義する人は PP/ST 作成者なので PP/ST 作成者については明記しない．なお，読み方については 2．5 で後述する．

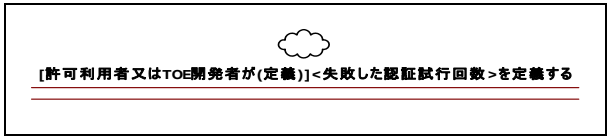


図 5 役割が記されているソフトゴール

③ パラメタが関わる場合について

“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．パラメタは，失敗した認証試行回数及び時間の閾値を含むが，それに限定されない．”（CC242 から転載）と記されていることから，このファミリでは失敗した認証試行回数と時間というパラメタが関わりあってくることがわかる．つまりソフトゴール名にこのパラメタとして使用されているものが関わる場合はそれを考慮する必要がある．今回はそれに関わるものを例えば図 6 のように< >で囲むことによって表現した．なお，読み方については 2．5 で後述する


図 6 パラメタを表すソフトゴール

### 2.4 CC からゴール木への作成例

手順ではまず FIA_AFL に記載された内容定義の文節をもとに転載したのち，それをもとにゴール木を作成する．その後，同様の文節に当たる付属の文節を付属書から転載した後，その付属書から転載された文節についてもゴール木を作成する．そして，前に作成した内容定義のゴール木と後に作成した付属書から作成したゴール木を連結させる．以下にゴール木を作成の一例を示す．

“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．”（CC243 から転載）

上記の要件をゴール木へと分解すると図 7 となる．以下に分解方法を示す

まず主語を見ると“FIA_AFL．1 認証失敗時の取り扱い”と書かれている．これは要求として扱えるので《FIA_AFL．1 認証失敗時の取り扱いの要求を定義する》という NFR ソフトゴールで定義する．さらにこれは主語なので以降の文章はこのソフトゴールのサブゴールになると考えられる．

次に，“利用者の不成功の認証試行が特定した数になった後”と書かれている．これは条件なので，条件文に関わる定義を用いる．このとき，パラメタである“利用者の不成功の認証試行が特定した数”と記されているのでこれも考慮すると《認証事象のリストに関して利用者の<特定した数の不成功認証試行>が生じたときを検出する》というソフトゴールを定義できる．このとき具体的にシステムが何をするべきか明確でないので NFR ソフトゴールとして定義する．最後に“セション確立プロセスを終了できることを要求する”と書かれているので《セション確立プロセスを終了する》という NFR ソフトゴールを定義した．

次に貢献関係を見る． “特定した数になった後，セション確立プロセスを終了”と記されているので手順を表す Satisficing①という貢献関係で《認証事象のリストに関して利用者の<特定した数の不成功認証試行>が生じたときを検出する》と《セション確立プロセスを終了する》をつなぐ．また，“できることを要求する”と記されていることから《FIA_AFL．1 認証失敗時の取り扱いの要求を定義する》際にこの 2 つの要求を必ず考慮する必要があると考えられるため，この 2 つをサブゴールとして貢献関係 AND でつなぐ．

Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

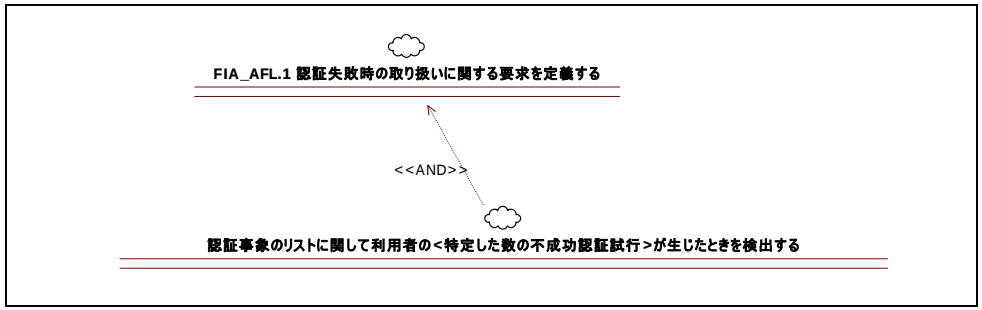


図 7　243 のゴール木

### 2.5 作成したゴール木の利用

作成したゴール木の読み方を説明する．上から見る以外は基本的には順番は関係ないが左のソフトゴールをルートとする部分木から見る．あるゴールをルートとするゴール木がある場合利用者はそのゴール木の枝をたどって具体的な値を定義する必要がある．このとき定義する要求をゴール木の階層毎に比較検討する．

- 権限を持つ役割による違い

基本的な読み方としては[権限をもつ役]が（何をする権限を持っているか）と読むことができる．たとえば図 5 は許可利用者あるいは TOE 開発者が失敗した認証試行回数を定義する権限がある．と読むことができる．

このとき，開発者が定義する場合はシステムにそのまま組み込むので新しい要件などは定義する必要はないが，許可利用者が利用する場合は利用者が操作を行うための権限なので新たに「変更する」などといったサービスが必要となる．

- パラメタとしての意識

不成功認証試行に関する値を定義する際にもし大きすぎる値を定義してしまうとシステムに与えるリスクは大きくなり，小さすぎるとシステムとしての利用性が悪くなる．同様に尺度，つまり無効にしている時間を定義する際に無効時間が無限大などであるとシステムは動くことができなくなるし，小さすぎると無効にする意味がなくなってしまう．つまりパラメタに関する要件を定義する際は，必ずパラメタを考慮しなくてはならない．

## 3. 作成したゴール木の有効性の検証

### 3.1 前提条件

検証方法の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，以下の条件があるとする．

まず前提条件として非機能要求である「WEB プラウザで十分な認証」という項目があるとする．この「十分な認証」について CC を適応させて分析した時，「認証失敗（FIA_AFL）」「秘密についての仕様（FIA_SOS(Specification of secrets)）」「利用者認証（FIA_UAU(User authentication)）」のセキュリティ要件が関係すると分析されたと仮定する．このとき秘密についてのゴール木を基に，秘密の検証（FIA_SOS．1）より「パスワードの作成方法」と「ID は口座番号である」を定義し利用者認証についてのゴール木を基に再認証(FIA_UAU．6)の要件より「認証は 2 回行う」を定義したとする．そして今回は認証失敗におけるセキュリティ要件を前提条件より「認証失敗」についての作成した FIA_AFL ゴール木から抽出する．これにより認証失敗したときに何をするべきかが把握可能となる．その後，実際に利用されているシステムに抽出した要件が適応されているか検証する．秘密の検証（SIA_SOS．1）と再認証(FIA_UAU．6)などの認証失敗以外の条件を適応させることにより認証失敗に関する前提条件を具体化させると以下のようになる．

I. 口座を WEB から見るシステムなので 2 回認証を行う
II. 認証は WEB プラウザで行う
III. 認証には ID（アカウント）として口座番号，パスワードとして 4 桁の数字と 8 桁以上の半角文字列両方を入力しなければならない
IV. 今回は個人認証のため認証のために用いる器具などはない
V. 操作は自宅のパソコンあるいは個人のパソコンから行われるものとする．つまり同じものを複数の人が共有するのではなく，同一の人物が利用する．

次に，実際に定義される要件として以下①～⑥の条件があると考えた．

① 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
② 認証失敗の尺度は時間経過でなく ID（アカウント）からの失敗経歴とする
③ 利用者が認証失敗を起こすと ID がロックされ，同時に管理者に ID がロックされたことが通知される
④ ロック解除のためには利用者が管理者に電話で連絡する必要がある
⑤ 管理者は利用者から連絡が来たらロックを解除する．このとき利用者のパスワードは初期設定のパスワードになっている
⑥ 今回，口座のパスワード認証ということでリスク管理の面から考えたとき認証失敗回数は管理者が定義しているのが妥当である

作成した FIA_AFL ゴール木に前提条件を付加させることにより上記で定義した実際に利用されている要件が定義可能か検証する．

### 3.2 検証

作成した FIA_AFL ゴール木を利用して要件①③ができないが定義可能か検証する．ゴール木の右側は操作に関するセキュリティ要件を表現しているのでこの右のゴールをルートとして持つ部分木をみる．すると《セション確立プロセス終了後のアクショ

Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

ンを定義する》の下位ゴールに《セション確立プロセスを無効にする》というゴールがあるのでこのゴールをルートとする部分木を図 8 にあらわす．このとき《　》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．

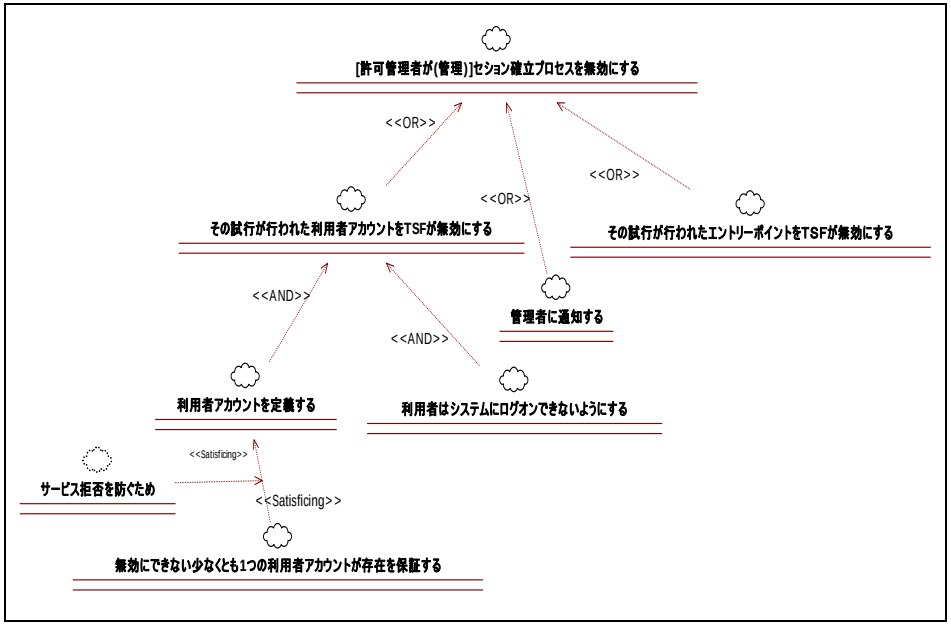


図 8　右側は操作に関するゴール木

《セション確立プロセスを無効にする》のサブゴールの貢献関係をみると《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》《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》の３つがある．このとき前提条件の『操作は自宅のパソコンあるいは個人のパソコンから行われるものとする．つまり同じものを複数の人が共有するのではなく，同一の人物が利用する』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と考えることができる．すると候補は《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》《管理者に通知する》となる．このとき，OR とは複数選択可能なので例えばどちらも選択するとする．更に《の試行が行われた利用者アカウントを TSF が無効にする》にはサブゴールがあるのでそのサブゴールを見ると《利用者アカウントを定義する》と《利用者はシステムにログオンできないようにする》の２つがある．このとき貢献関係は AND 関係なのでどちらも選択する必要がある．まず《利用者アカウントを定義する》のサブゴールをみると《無効にできない少なくとも 1 つの利用者アカウントが存在を保証する》という要件がある．このとき理由ソフトゴールには《サービス拒否を防ぐため》と書かれている．つまり，これは「サービス拒否を防ぐため無効にできない少なくとも 1 つの利用者アカウントが存在を保証する」という要件を定義している．

以上からゴール木と前提条件から《管理者に通知する》《利用者はシステムにログオンできないようにする》《無効にできない少なくとも 1 つの利用者アカウントが存在を保証する》という要件を定義できた．このとき要件①③を確認すると，まず要件①の『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』については《利用者はシステムにログオンできないようにする》という要件を定義することで定義できたと考えた．次に③の「利用者が認証失敗を起こすと ID がロックされ，同時に管理者に ID がロックされたことが通知される」については《管理者に通知する》《利用者はシステムにログオンできないようにする》という２つの要件を定義することで定義できたと考えた．更に，今回ゴール木を用いることで要件では考え付かなかった「サービス拒否を防ぐため無効にできない少なくとも 1 つの利用者アカウントが存在を保証する」という要件も今回作成したゴール木から定義できた．

## 4. 考察

セキュリティを熟知してないシステム開発者でもセキュリティ要件を定義可能とするような類似の目的を持つ関連研究[6]がある．関連研究では CC の要件をデータベースとしてセキュリティを熟知しているシステム開発者があらかじめ定義していた．これではなぜそのセキュリティ要件を採用したのか，また CC からどの要件を採用したのかが不明瞭であった．更に 2．1 節でも述べたが，CC はエレメントをシステムに合う具体的な値に書き換える必要があるが，関連研究では予め書き換えて定義している部分と，書き換えていない部分やエレメント以外は示していなかった．これでは不成功認証試行回数の値を定義できる人や無効にする尺度の定義や有効にする条件などを考慮することができない．そこで，今回ゴール木により目的から手段を表現することで，その要件の選択した根拠を明確にすることや追跡することが可能となる．そして，CC に記されていることを全て木にしたので網羅的に要件を定義することが可能となる．更に，このゴール木に知識を蓄えていくことで，再利用可能となる部分もでてくると考えた．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でもゴール木を定義することができるのではないかと考えた．

また，今回参考文献[5]を利用したが全てのルールを利用したわけではない．その理由としてガイドラインはいくつかの要件が箇条書きに記されているのに対して，CC は文章により要件が定義されていたという点やガイドラインは値の定義などが記され

Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

ていなかったのに対して CC は要件なので属性と操作に分けて考える必要があった点，そしてガイドラインは要件を１つ１つ定義しているのに対して CC は付属書の注釈のように一つの要件を違う言い回しで書いているという点があると考えた．

そして CC を対象としたが，分析の際に定義の要件と付属書の要件の繋がり，抽象的な単語（例えば，認証事象のリスト）の意味，例として記述されている要件の対象の把握や例と必要な要件の区別，そして管理・監査の要件の定義方法が困難であると感じた．また，役割の権限毎の違いによりどのような可能性があるか例示などがあると良いと感じた．そして，上に記された要件に対してその下に同じ内容をもう少し具体的に記しているという形や，あるいは具体的な注釈を付けくわえているので全体的な流れの把握が困難であると感じた．

## 5. まとめ

今回我々は CC という仕様書に定義されたセキュリティ要件というものに対してゴール木を用いることでセキュリティを熟知してないシステム開発者でもセキュリティ要件を定義可能であると提案した．ゴール木を作成することにより要件を満たしているか網羅的に確認することやその要件を定義した根拠を捉える事が可能となる．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でもやるべきことをある程度までなら考慮することができると考えた．また，今後の展望としてゴール木をデータベースとして扱うことでセキュリティを熟知していないシステム開発者でも過去の事例を参照して要件を定義できるようなツールを作成したいと考えている．これによりセキュリティをシステムに取り込む際になぜこの要件を満たす必要があるのか，あるいはなぜこの要件を満たす必要がないのかという根拠ができると考えた．さらに定義したセキュリティ要件をデータベースに記録することでセキュリティを熟知していないシステム開発者でも過去の要件を参照,あるいは再利用することで定義可能と考えた．

今後の課題としては，今回は考えなかった管理についての要件や監査についての要件をゴール木で定義して作成したゴール木とのつながりの把握をする．また，作成したゴール木の妥当性の検証を今度は自分ではなく他の人にも実験してもらう．更に今回定義したルールのみで他の要件すべてが定義可能かわかりやすい識別と認証クラスに定義されたファミリ全てをゴール木で表す．また，今回ゴール木を作成するに当たり Satisficing という貢献関係が 2 つの意味をもってしまったのでそれについても考慮する必要がある．更に，後に CC とは別に定義した要件にパラメタが出てきたときにパラメタ表現するにはゴール木の名前を変える必要などが出てくるのでそれについても考慮するべきである．

## 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 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
[2] i*homepage．http://www．cs．toronto．edu/km/istar
[3] L．chung，B．Nixon，E．yu，and J．Mylopoulos，Non-Functional Requirements in Software Engineering．Academic Publishers，1999．
[4] Non-Functional Requirements Modeling Tool
http://www．utdallas．edu/~supakkul/tools/softgoal-profile/softgoal-profile．html
[5] 山本，神戸，電子タグプライバシー保護ガイドラインのゴール分析，電子情報処理学会信学技報 ，pp25-30，2006.
[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)

Vol.2009-SE-164 No.10
Vol.2009-EMB-13 No.10
Vol.2009-CSEC-45 No.10
2009/5/28

# 付録

## 付録 A. 1 作成したゴール木全体