文脈に基づくセキュリティモデルの強制系の実現と評価
Design and Implementation of a Context-based Security Model

飯島 正 †, 城戸 聡 †
Tadashi IIJIMA†, and Satoshi KIDO†
† 慶應義塾大学　理工学部
†Faculty of Science and Technology, Keio Univ.

要旨
文脈を意識したアクセス制御ポリシーの強制系の設計に関して報告する．このアクセス制御モデルではオブジェクト指向ペトリネットを文脈表現として採用している．オブジェクト指向ペトリネットには幾つかのバリエーションがあるが，本報告では，nets-within-nets 意味論と呼ばれる意味論に基づくものを採用する．特に，異なる管理組織にまたがったワークフローにおける動的なアクセス制御モデルに利用することを目的としている．全体的なアーキテクチャは XACML の構成を参考にしており，セキュリティポリシーの記述には Scala による内部 DSL を利用している．

1. はじめに

文脈を意識したアクセス制御ポリシーの強制系の設計に関して報告する．このアクセス制御モデルではオブジェクト指向ペトリネットを文脈表現として採用している．特に，複数のデータ管理組織にまたがって分散されている情報にアクセスすることを想定している．従来，単一のデータベースにアクセスするのであれば，その Facade としてのデータベース管理システムにアクセス制御モデルを埋め込んでおくことで，個々のアプリケーションソフトウェアにアクセス制御モデルを作りこむ必要はなかったが，近年，それを前提とすることは必ずしも有効ではなくなってきている．そこで，分散した複数の情報資源へのアクセス制御モデルをアプリケーションソフトウェアに作りこむのではなく，抽象化し外部化することで，共通的なフレームワークを構築することを試みている．もちろん，アプリケーションソフトウェア側だけでは，アクセス制御ポリシーを強制することはできないので，情報源の側とアプリケーションソフトウェアの側で協調してポリシーを順守したアクセス制御を成立させることになる．

本報告で利用するオブジェクト指向ペトリネットは，更にオブジェクト指向の考え方に基づくモジュール性をとり入れており，組織にまたがるワークフロー，組織ごとのワークフロー，情報にアクセスする主体 (アクター) 毎の状態遷移，アクセスされる資源の状態遷移などのそれぞれに対して，個別にプロセス表現を与えて協調動作させることができるモデルである．この点が，前述した「複数のデータ管理組織にまたがって分散されている情報にアクセスする」際の文脈表現という目的に合致していることから，本報告では，このオブジェクト指向ペトリネットを文脈表現に採用することを提案する．

次節では，本報告で想定している具体的事例から，文脈を意識したアクセス制御 (Context-Aware Access Control: CxAC と略す) の必要性を示す．続く，第 3 節では，net-within-net 意味論に基づくオブジェクト指向ペトリネットについて解説する．それを踏まえ，第 4 節では，CxAC の一つのサブクラスとして，ロールに基づくアクセス制御モデル (Role-Based Access Control: RBAC) [2] に文脈をとり入れた CxRBAC を示す．さらに，第 5 節においてポリシー強制系の設計に関して述べる．

2. 具体的な想定事例

事例として医療情報へのアクセスに関して検討する．一つ目の想定事例としては，状況に応じた動的なアクセス権限の変更を取り上げる．近年，救急隊員が搬送中の患者に対して救命処置にとって，必要な情報を有効に取得することの必要性が認識されてきている．まず，身元確認が最初の手がかりとなるが，そのうえで，既往歴，アレルギーの有無，医療禁忌情報，服薬中の薬の情報などが重要である．これは現場や搬送中における救急隊員の判断のためばかりではなく，救急隊員が医師の指示を仰ぐ場合にも，また搬送先の病院に到着後に速やかに治療行為に着手するためにも有益な情報である．しかし，救急隊員にとって，こうした情報へのアクセスは，緊急時以外には許諾されるものではない．また，複数の病院，臨床検査会社，薬局に分散している可能性もあり，緊急性が求められるにも関わらず，情報入手のプロセスは複雑である．ここで求められているのは，以下の 3 点である．

- 患者の生命にかかわる緊急時に，広範な医療情報に迅速にアクセスできる

- 緊急時に必要に応じて，アクセス権限を動的に拡大できる
- 複雑な情報入手プロセスをワークフローとして整備することで，緊急時に意識することなく統合された情報にアクセスできる

また，別の想定事例として，医療情報の共有化を取り上げる．近年，カルテの電子化が進む一方で，情報の共有化は必ずしも進んでいない．単一の医療機関だけで診断を受けるのではなく，患者自身がよりよい治療法を選択するという発想から，セカンド・オピニオンを求めて他の医療機関で医師の意見を求めるという行為も従来より増えてきており，専門のセカンドセカンドオピニオン外来を設置している医療機関もある．但し，セカンドオピニオン外来では，治療だけでなく検査も行わず参考意見という形で相談ができるだけということが一般的なので，元々の主治医から，検査データや診察記録・処置記録などの医療情報の提供を受ける必要がある．この際，患者本人が閲覧を許可したとしても，現状では電子的な医療情報の流通の基盤整備は進んでいないために，画像検査データの一部を除けば紙媒体での提供が主体である．そこには電子カルテシステム間の相互互換性，各医療機関ごとの情報管理ポリシーの違いといった障壁がある．しかし，閲覧や情報の読み出しだけであれば，適切なアクセス制御のもと，ネットワークを介した情報流通の可能性はありうる．ここでのアクセス制御のためには，患者本人の権利に加えて，医療情報を生成した医療機関のもつ権利も合わせて配慮する必要がある．

さらに，患者の転居に伴う転院や，脳卒中の急性期から回復期，維持期への移行に伴う転院の際に，複数医療機関の医療連携体制が必ずしも確立されておらず，同じ検査が繰り返されたりすることがあり医療費負担も増してしまうことがある．医療機関間だけでなく，臨床検査会社で保有している検査データも，被検者本人の許諾によって流通させることは，決して悪いことではない．これは，検査データやカルテは本来，被検者/受診者本人に帰属すべきものであるという見解につながる．加えて，体重や血圧などの健康情報を日常的に蓄積する個人健康情報 (Personal Health record; PHR) との連携もありうる．

このように近年のアクセス制御のモデルは，従来，特定組織内の集中管理されたデータベースにアクセスするケースのように，Facade としてのデータベース管理システムにアクセス制御モデルを持たせておくだけでは不十分である．異なるデータ管理組織に分散された情報に対し，状況を意識したアクセス制御 (Context-Aware Access Control: CxAC と略す) を実現しなければならない．

3. オブジェクト指向ペトリネット

3.1. オブジェクト指向ペトリネットとは

本研究では，文脈表現としてオブジェクト指向ペトリネットを用いている．オブジェクト指向ペトリネットとはオブジェクト指向概念に基づいてモジュール性を取り入れたペトリネットの拡張モデルであり，多くのものが提案されている．ここでは，nets-in-nets 意味論に基づく参照ネット (Reference Net) [1] の考え方を採用したオブジェクト指向ペトリネットを採用している．

参照ネットにおいては，トークンには 2 通りある．一つは，通常の P/T ネット (Place/Transition ネット) におけるトークンである単純トークン (Black Token) とよぶ．プレースに存在する単純トークンは，トークン数で示す．もう一つは，別のサブシステムを表現するサブネットへの参照 (reference) に相当する参照トークン (Reference Token) である．

プレースは，単純トークン用の単純プレースと，参照トークン用の参照プレースに分類できる．

- 単純トークン (Black Token)[1]
- 参照トークン (Reference Token)

ここで，参照トークンが参照しているサブネットは，オブジェクト・ネットと呼ばれ，一つのオブジェクトとしてのアイデンティティをもつ．参照トークンも，単純トークンと同様，ネットワーク中を遷移することができるが，一つの参照トークンがトランジションの発火によってコピーされることもある．その際には，あくまで共通のオブジェクトネットへの参照がコピーされる点に注意されたい．

後述する (図 1 左) ように，各トランジションの発火の際には，そのトランジションと同期する (interaction 関係にある) ペトリネットにおいては，同名のメソッドが実行される．すなわち，そうした同期関係 (interaction 関係にある) ペトリネットが実行主体 (アクター) として位置づけられることになる．受

[1]慣習的にブラック・トークンと呼ばれるが，ここでは分かり易さのために単純トークンと呼ぶ

け渡しされるパラメータは，基本的に，そのトランジションへの入力アーク上の weight で表現される．それに加えて，同期関係にある他のネットから受け渡される参照が付け加えられる．それらは，?(入力パラメータ) ならびに !(出力パラメータ) として受け渡される．

3.2. オブジェクト指向ペトリネット OPeN

以下では，具体的なオブジェクト指向ペトリネットのモデルとして，筆者の主宰する研究室で継続的に開発を行ってきたオブジェクト指向ペトリネット OPeN ファミリー (the Object-oriented Petri Net family) の中でも，特に人と人の連携して行う協調作業や，SOA におけるサービス間連携を記述するためのビジネスモデル記述 (ワークフロー) に特化した OPeN/WF を用いている．

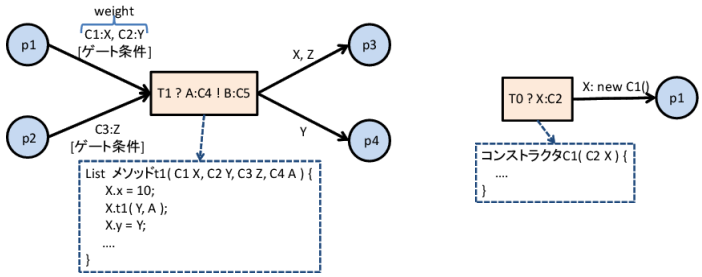


図 1: OPeN におけるアークとトランジション

OPeN では，個々のオブジェクト・ネットは，クラス記述から生成されるインスタンスに相当する．クラス記述は，オブジェクトを定義するものであり，本稿執筆時点では，opn(もしくは従来からの慣習から pnmlx) という拡張子を持つ XML ファイルとして記述する．クラスが定義するオブジェクト・ネットは，そのオブジェクトの振る舞い (ライフサイクル) を記述する一つのペトリネット，そのオブジェクトが持つプロパティ(インスタンス変数) 群と，ライフサイクルを遷移する際に使われるメソッド群である．メソッドは，主に，トランジションの発火条件であるゲート条件の記述や，トランジション発火時に起動されてプロパティの値の更新に使われる．図 1 に OPeN におけるアークとトランジションのアノテーションを示す．

トランジション t_1 の発火によって，同じ名前をもつメソッド t1 が起動される[2]．現時点のプロトタイプ仕様では，プロパティの型やメソッドの定義のための言語仕様は，厳密には定義していないが，図 1 に準拠し，基本的なリフレクション機能を備えた一般的なオブジェクト指向言語 (たとえば Java や Scala) のクラス定義 (ネットと同名のクラス定義) に外部化しており，相互に対応付けている．

複数のオブジェクト・ネットから構成されるシステムは階層的に表現されるが，これには，プロジェクトという単位を用いる．しかし，ここでは，まず分かり易さを優先して多階層の構造ではなく，2 階層の構造に限定して，その意味論を概念的に説明する．これは，[1] における EOS(初等オブジェクトシステム) に相当する．2 階層の構造では，全体的な振る舞いを統制するオブジェクト (ペトリネット) と，それに規定された振る舞いを行うサブオブジェクト (ペトリネット) 群から構成される．ここでは，全体的な統制を記述する方を EOS の慣習に従いシステム・ネットと呼び，統制されたサブオブジェクトの側をオブジェクト・ネットと呼んで説明することにする[3]．

システム・ネットとオブジェクト・ネットの間では，一部のトランジションの発火が同期的に行われる．システム・ネットとオブジェクト・ネットの同期関係は，双方のトランジションの対の集合として表現される．システム・ネットにおいて，あるトランジション T が発火するためには，以下の三条件が成り立たねばならない[4]．

(a) システム・ネットにおいて，単純トークン B_i に関して発火条件を満たしていること，
(b) システム・ネットにおいて，参照トークン R_j に関して発火条件を満たしていること，
(c) T の発火に寄与している参照トークン R_j が参照しているオブジェクト・ネット中で，T と同期関係にあるトランジションが発火可能であること．

[2]後述するオブジェクト指向ペトリネットの発火則「Interaction(相互作用)」により，t_1 と同期するトランジション t_{s1} が，このペトリネットのサブネット中に存在し，かつそれが発火可能であれば，その t_{s1} も同時に発火する．t_{s1} が発火可能でなければ t_1 も発火できない．トランジションの入力アークがメソッドの入力パラメータに対応する．

[3]階層的なシステムの場合には，ここでいうオブジェクト・ネットもまた，より下位のサブペトリネットからみれば相対的にシステム・ネットに相当する．OPeN は EOS の概念を多階層に拡張しているが，混乱のない限り，最上位ネットに限らず，隣り合う階層のネット間の関係を，システムネットとオブジェクト・ネットというように称することとする

[4]但し，OPeN の場合には，二階層の EOS ではなく，多階層を許容しているので，条件 (c) は再帰的に，より下位層にあるオブジェクト・ネットに対して適用されていくことになる．

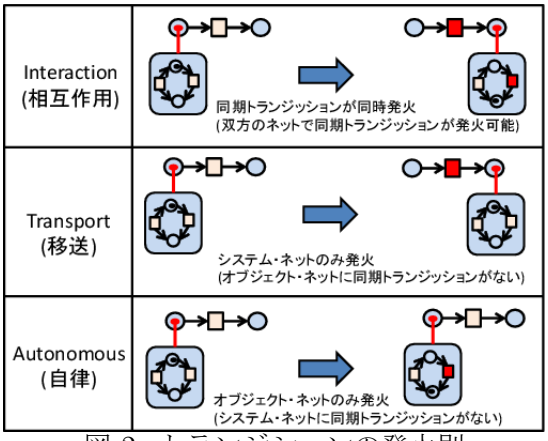


図 2: トランジションの発火則

こうした発火のメカニズムから，nets-within-nets 意味論 (ないし参照意味論)，およびその拡張に基づくオブジェクト指向ペトリネットにおいては，(a) interaction(相互作用), (b) transport(移送), (c) autonomous(自律)，という 3 通りの発火則が考えられる．まず，システム・ネットとオブジェクト・ネットが同期して遷移するタイプであり，interaction(相互作用) と呼ばれる．次にシステム・ネットの発火に際し同期して発火するオブジェクト・ネットのないタイプがあり，transport(移送) と呼ばれる．これは，オブジェクト・ネット内部には何の状態遷移も起こっていないまま，システム・ネット上にある参照トークンが移動する様子が，モバイル・エージェントの移動に対応づけられることから名づけられている．最後に，システム・ネットとは独立に，オブジェクト・ネット内のトランジションが発火するものであり，autonomous(自律) と呼ばれる．その呼称は，システム・ネットに制御されることなく，オブジェクト・ネット内部で自律的にトークンの遷移（すなわちそのオブジェクト・ネットの状態遷移）が引き起こされていることに由来する (図 2).

4. 文脈指向ロールベースアクセス制御モデル

文脈を意識したアクセス制御モデル (CxAC) には，そのベースとするアクセス制御モデルと，具体的にどのような状況を取り扱うかで，2 つの観点からサブクラスが考えられる．ここでは，ベースとするアクセス制御モデルに RBAC(Role-Based Access Control) を採用した文脈指向ロールベースアクセス制御モデルを提案する．

一般に，アクセス権限は，主体–操作–対象の三つ組のうち，許可されているものの集合で表現できる．ロールベースアクセス制御モデルは，情報にアクセスする主体ごとにアクセス権限を定義するのではなく，ロールすなわち役割に基づいて主体を分類し，ロールごとにアクセス権限を定義するものであり，現在，広く使われている．ロールに分類することで大幅にアクセス権限の定義数を減らすことができる．電子カルテの場合，病院職員の一人一人にアクセス権限を定義するのではなく，医師や看護師，薬剤師，検査技師，事務員といった役割ごとに，アクセス権限を定義するものである．

さらにロールは，継承階層に構造化することで，より体系化し記述量を減らすことができ，ポリシーの保守性も向上する．たとえば，医師や看護師を診療科毎にグループ化したり，部長や主任といった役職ごとにグループ化することが可能である．そうした分類観点は複数直交するものも考えられるので，単純な木構造の継承階層を基本とするとしても，複数のロールを MixIn させて，より複雑な構造を持たせることもありうる．たとえば，診療科ごとに認定専門医という資格の有無に基づく分類や，特定の患者に対しては担当医か否かといった分類がありうるが，ここまで細分化した場合には「属性に基づくアクセス制御モデル」と考えた方がよい．

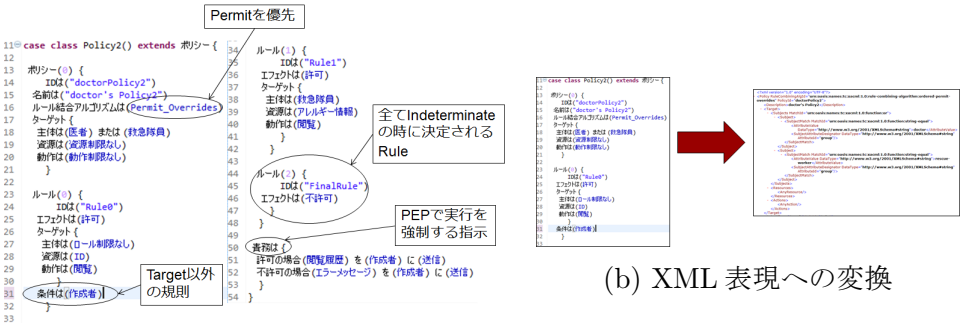
また，一般にこうしたロール割り当ては，アドホックに変更されるものではない．すなわち，医師や看護師といった役割はそう簡単に相互に変更されるものではなく，役職に関しても変更 (昇格，降格) はありうるが，アドホックな変更とは言い難い．したがって，緊急時のみのアクセス権限の拡大といった状況に応じた動的変更機構は，通常は，スパンの長いロール割り当てとは別に扱われる．

しかし，ここでは，アドホックな動的変更もロール毎のポリシーの合成として表現し，

操作–対象–判定–条件–責務

の組の集合を粒度の様々なロール毎にポリシーとして与える．ロールごとのポリシーの多重的な MixIn による合成 (順序がある) を動的に行うことによって権限を確定させるものとする．判定の部分は，許可/拒否/未定の 3 通りのいずれかである．操作と対象と判定が一致しているポリシー同士は，条件は論理積で，責務ないし義務 (obligation) は集合和によって合成される．条件の一部として文脈を使うことができる．ここでの文脈は，主体側と対象側の両方にありうる．

ポリシーの記述には，プログラミング言語 Scala の内部 DSL(Domain Specific Language; ドメイン固有言語もしくはドメイン特化言語) 表現を利用することができ，XACML で標準的に用いている XML 表現に変換することができる (図 4).

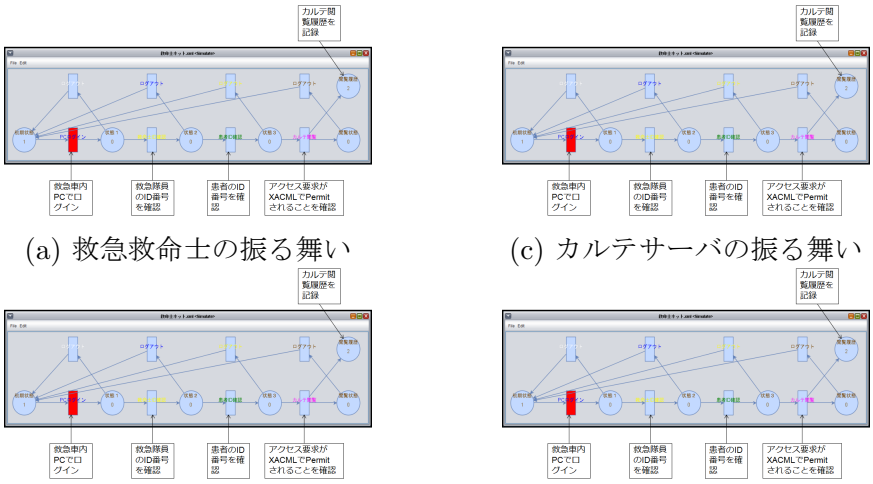


(a) DSL 表現　　(b) XML 表現への変換

図 3: ポリシーの DSL 表現と XML 表現への変換

アクセス者と情報資源 (もしくはそのサーバ) のそれぞれの振る舞いをそれぞれオブジェクト指向ペトリネットで表現し，協調動作はトランジション間の同期によって表現する．さらに，強制すべきポリシーをオブジェクト指向ペトリネットに変換し，他のペトリネットと同期関係を定義することによって，ポリシー強制系 (PEP=Policy Enforcement Point) は，アクセス者にポリシーに従った振る舞い (たとえば責務の遂行) を強制する．

以下に簡単な例を示す．この例では，救急車の中で救急救命士が，所持している IC カードを使って，医療センターのカルテサーバに保管されている患者の医療情報（たとえばアレルギー情報など緊急性の高いもの）にアクセスする際のポリシーを与えている．現行では，特定行為を除き，医師による具体的指示 (オンラインメディカルコントロール) がなければ，救急救命士による医療行為は禁じられている．しかし，アナフィラキー症状で生命が危険な状態にあるとき，その症状を一時的に緩和し医師の治療を受けるまでの間ショックを防ぐために，予めエピペン (アドレナリン自己注射薬) を処方されている場合に限り，救命士は，代行注射することもできるようになっている [5] .



(a) 救急救命士の振る舞い　　(c) カルテサーバの振る舞い

(b) 救命士 ID サーバの振る舞い　　(d) 強制すべきポリシー

図 4: ポリシーの DSL 表現と XML 表現への変換

オブジェクト指向ペトリネットでは，オブジェクト・ネットで個別の要素の振る舞いを表現し，システム・ネットでそれらの協調の様子を表現することができる．2 階層以上を持つことによって複数組織にまたがったワークフローと，個別の組織のみならず，その中を動くアクター（情報にアクセスする主体）と，情報資源（アクセスされる対象）の個別の振る舞い（状態遷移）を表現することができる．ペトリネットにおいて状態はマーキングとして扱うことができる．したがって，状態の時系列すなわち状況は，指定したいマーキングの時系列を作り出すペトリネット（ワークフローのサブネット）で表現することができる．そこで，本報告では，トランジションの発火系列で文脈を表現するものとする．

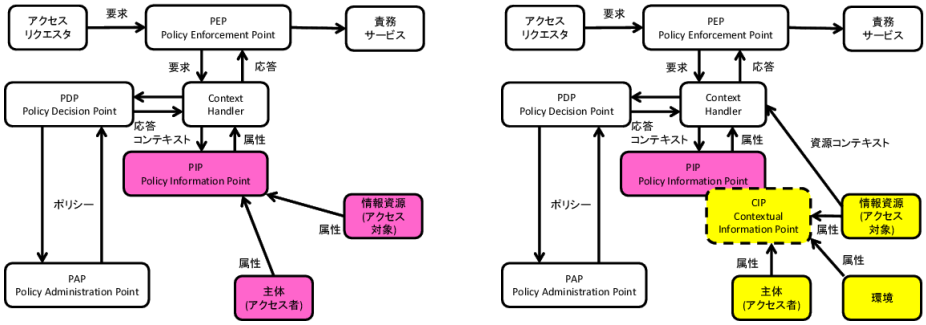
情報資源を意味するオブジェクトネットと，アクセス者である主体を意味するオブジェクトネットの間では，両者のトランジション間で同期発火が発生する際に情報アクセスが行われる．文脈はワークフローのサブペトリネットで可能な発火系列の集合で表現し，これを文脈パターンと呼ぶ[5]．ある発火系列が文脈パターンにマッチするとは，それが，そのサブペトリネットで可能な発火系列の集合の要素であ

[5]あるペトリネット n のサブペトリネット $sub(n)$ は n のトランジション集合，ペトリネット集合，アーク集合の部分集合で構成できるペトリネットである

る場合に限る．

5. アーキテクチャ

ここで想定しているアーキテクチャは，OASIS により標準化作業が進められている XACML(eXtensible Access Control Markup Language) [3] のアーキテクチャの拡張である．最新の XACML3.0 に対応していないが，Sun Microsystems の XACML 実装 [4] を元に実装を行った．



(a) XACML　　(b) その拡張

図 5: XACML のアーキテクチャとその拡張

XACML のデータフローモデルには，コンテキストハンドラ，リソース・コンテキスト，応答コンテキストといったキーワードが含まれているが，本研究における文脈表現が取り扱われるのはそこではない．権限の判定を行う PDP(Polocy Decision Point) および，アクセス制御を実施する PEP(Policy Enforcement Point) を結びつけるコンテキストハンドラの前段として，主体や情報資源及び環境の属性情報を取り扱う PIP(Policy Information Point) の部分を拡張することになる (Contextual Information Point=CIP と名付ける)．図 5 における点線で描かれたモジュールが拡張点である．

「文脈情報の偽造/なりすまし」を避けるためには．文脈情報の正当性を保証することが必要である．しかし，今回の提案モデルは，管理者の異なる分散した情報資源を前提としている．そこで，アクセス主体の文脈情報や他の管理組織のもとにある情報資源の文脈情報の保証のためには，外部の信頼できる第三者による承認をもって行うものとする．

6. まとめ

文脈を意識したアクセス制御ポリシーの強制系の設計に関して報告した．このアクセス制御モデルではオブジェクト指向ペトリネットを文脈表現として採用している．全体的なアーキテクチャは XACML の構成を参考にしており，セキュリティポリシーの記述には Scala による内部 DSL を利用している．但し，責務のための DSL 表現は暫定的なものであり，今後，具体的な記述例を増やしながら改良していく予定である．ポリシー自体も責務を含め，オブジェクト指向ペトリネットに変換し，アクセス者やアクセス対象の資源（もしくはそのサーバ）のペトリネットの特定のトランジションとの間で同期関係を規定することによって，ポリシー強制系 (PEP) は，そのポリシーの強制を行う．

謝辞

当研究室・Sec 班 (セキュリティグループ) のこれまでの卒業生の皆さん (塩田 哲哉 (2012 年度学部卒)，山上 燦 (2011 年度学部卒)，雨宮 美奈帆 (2010 年度学部卒)，山本 由香里 (2009 年度学部卒)，奥村 恭平 (2008 年度学部卒)) に感謝します．

参考文献

[1] Rüdiger Valk: “Object Petri nets? Using the nets-within-nets paradigm, ” LNCS 3098, pp.819–848, Spriner, 2004.
[2] R. S. Sandhu, E. J. Coyne, et al. : “Role-based access control models, ”IEEE Computer, vol.29, no.2, pp.38–47, 1996.
[3] OASIS Standard : “eXtensible Access Control Markup Language(XACML) 3.0,” 22 January 2013, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf
[4] Sun Microsystems, Inc.(当時): “Sun’s XACML Implementation,” 16 July 2004, http://sunxacml.sourceforge.net/
[5] 厚生労働省医政局指導課長: “「救急救命処置の範囲等について」の一部改正について,” 医政指発第 0302001 号, 平成 21 年 3 月 2 日, http://www.mhlw.go.jp/topics/2009/03/dl/tp0306-3a.pdf