文脈に基づくアクセス制御の応答性の評価

Evaluation of responsiveness of access and authorization control based on Context-Aware

尾崎稜太 †, 飯島正 ‡
Ryota Ozaki†, and Tadashi Iijima‡
† 慶應義塾大学 大学院 理工学研究科 開放環境科学専攻
‡ 慶應義塾大学 理工学部
†Graduate School of Science and Technology for Open and Environmental Systems, Keio Univ.
‡Facalty of Science and Technology，Keio Univ.

要旨

近年では，個人情報であっても複数の組織にまたがって管理されており，アクセス制御サーバを外部化し，多様な連携利用を可能とすることが求められている．そうした複数の組織で管理される個人情報には高度な機密性や，アクセスに緊急性が求められるケースも想定される．そうした情報へのアクセスの利便性を保ちつつ適切なアクセス制御を行わなければならない．

そのため，本研究プロジェクトではオブジェクト指向ペトリネットによるワークフローを文脈とし，それに基づいてアクセス制御を行う，文脈指向ロールベースアクセス制御モデル (CxAC) を提案し，アクセス制御サービスを実現してきた．

本報告では事例研究を通して，文脈を意識したアクセス制御の有用性を示すとともに，応答性に対する定量的評価を行い実用可能性を確認した結果を報告する．

1. 研究背景

近年，ネットワーク技術の発達に伴い，SNS などが普及し，インターネット上に誰でも個人情報を作成することができるようになった．しかし，これらの個人情報は様々なユーザにより閲覧される可能性がある．つまり，個人情報が見知らぬユーザに悪用されてしまう危険性が大きくなっている．例えば，医療施設では，従来は紙で作成されていたカルテや処方箋が電子化されてきている．紙のカルテであれば，病院内での管理を適切に行えば外部に流出することはなかったが，電子化されたカルテであると，ロールに応じたアクセス可否の設定など，細かいアクセス制御が必要となる．そこで，ロールベースアクセス制御 [1] などのアクセス制御モデルが提案されてきた．しかし，その多くが近年のアクセス環境の変化に対応しきれていないのが現状である．

本研究プロジェクトにおけるアクセス制御モデルでは，特に複数の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．従来はあるデータベースにアクセスするとき，そのデータベース管理システムにアクセス制御モデルを組み込むことでアクセス制御を実現していたが，今後はアプリケーションソフトウェアが複数のデータ管理組織にアクセスし，組織をまたがった情報の移動が行われる状況が想定される．そのアクセスに対して制御を行う際には，個々のデータベース管理システムでのアクセス制御ではなく，外部化されたシステム上でのアクセス制御が求められる．また近年，ウェアラブル端末の普及や GPS の高性能化に伴って，利用者の位置情報や時間情報などを文脈とした文脈に基づくアクセス制御が注目されてきている．

そこで，本研究プロジェクトではアクセス制御モデルの共通のフレームワークを構築することを目的とした，オブジェクト指向ペトリネット [2] を文脈表現とした，文脈に基づいたセキュリティモデルを提案してきた．オブジェクト指向ペトリネットとは，組織ごとのワークフロー，組織にまたがるワークフロー．情報にアクセスする主体ごとの状態遷移，アクセスされる資源ごとの状態遷移など，それぞれに対して個々にプロセス表現を与えることができるモデルである．この点で，前述の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想定にあてはまることから，本研究プロジェクト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．

共通のフレームワークとしてのアクセス制御では，電子化情報への「読み・書き」の操作に対する制御だけではなく，機器に対してどのような操作を許可するかの規定もアクセス制御のフレームワークに含めることが求められる．そこで，本研究プロジェクトでは，文脈を考慮したアクセス制御である文脈指向ロールベースアクセス制御モデル (CxAC:Context-aware Access Control) を提案し，このモデルを

適用することで，文脈に応じてアクセス権限を動的に変更できること，情報の加工度合いに応じたアクセス制御も可能であることを示す.

また，外部化されたシステム上でのアクセス制御を想定しているため，多くのアクセス権限への問い合わせが同時にあった場合にも適切なサービスを行えることが求められる. そこで，本論文ではこのセキュリティモデルにアクセスを集中させた場合にも遅れることなく結果を返すことができることを示す.

2. 提案

本研究プロジェクトでは，文脈指向ロールベースアクセス制御（CxAC : Context-aware Access Control）を提案している. 文脈に基づいたセキュリティモデルには，そのベースとするアクセス制御モデルと，状況判断の方法とで，2 つの点からサブクラスを考えることができる. 本研究では，ベースとするアクセス制御モデルとして RBAC(Role-Based Access Control) [1] を採用し，状況判断にはオブジェクト指向ペトリネット [2] によって表現されるワークフローを用いる.

2.1.RBAC ベースのアクセス制御

アクセス制御は，一般に，主体・対象・操作の 3 つの組のうち，許可されている組の集合で制御される. 特に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にアクセス権限を設定するのでなく，ロールに対してアクセス権限を設定し，各主体をそのロールにそれぞれ割り当てるものである. ロールごとにアクセス権限を設定することで. アクセス権限の定義数を大幅に減らすことができる. 例えば，病院の場合であれば，従業員それぞれに権限を設定するのではなく，医者や看護師のようなロールごとにアクセス権限を設定する.

2.2. 文脈に基づく状況判断

一般的に，ロールの割り当ては頻繁に起こるものではなく，医師や看護師といったロールが相互に変更されることは少ない. 役職に関しては変更は起こるが，頻繁に発生することではない. したがって，緊急時のアクセス権限の拡張などの状況に応じた動的な変更は，このようなロール割り当てとは別に扱われる. しかし，ここでは動的な変更もロールごとのポリシの一部として扱い，

操作—対象（資源）—判定—条件

の組の集合をロールごとにポリシとして与える.

3. 設計と実装

3.1. オブジェクト指向ペトリネットによるワークフロー表現

ペトリネットは，グラフ上に円または楕円で描かれ条件を示すプレースと，長方形で描かれイベントを示すトランジションと，矢印で描かれ条件とイベントの関係を示すアークと，点で描かれそれがプレースの上にあることで条件の成立を示すトークンで構成されている.1 つのプレース内に複数のトークンを置くことも可能である．トランジションで示されたイベントが発生することをトランジションが発火するという.

電話を例にペトリネットを用いて簡潔にモデル化したものが以下の図 1 であり，トランジション t1 は「電話をかける」というイベント，t2 は「話をする」というイベント，t3 は「電話を切る」というイベントを表している．また，初期状態として電話をかける前の状態にトークンを置いた. この場合，電話をかけるというイベントを行わないと，話をするイベントや，電話を切るというイベントが実行できないということが表現できている.

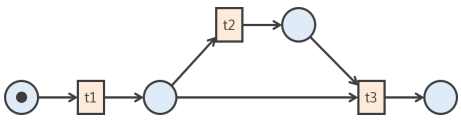

図 1: ペトリネットの例

CxAC は，ペトリネットを拡張させたオブジェクト指向ペトリネットを用いている．トークン自身にさらにペトリネットを持たせることができるため，複数の組織をまたがったワークフローのシステムネットと，個々の組織のワークフローであるサブネットを表すことができる．アクセスの主体と情報資源のそれぞれの行動 (ワークフロー) を，オブジェクト指向ペトリネットで表し，同時に動作を行う場合はトランジション間の同期によって表す．また，強制すべきポリシをオブジェクト指向ペトリネットに変換し，ほかのペトリネットとの同期関係を設定することによって，アクセス者にポリシに沿った行動を強制する．例えば，患者が他の病院から移転してきた際にも，患者や情報資源のワークフローと医療機関をまたがったワークフローから状態遷移を取得し，その情報からアクセス制御を行うことが可能である．オブジェクト指向ペトリネットによるワークフローの例を図 2 に示す．この例では，病院間ネット (システムネット) と電子カルテネット (オブジェクトネット) の一部のトランジションを同期させているため，編集中はトークンが t2 が発火されることで移動してしまい，t1 が発火できなくなるため，電子カルテの病院間での移動ができなくなる．

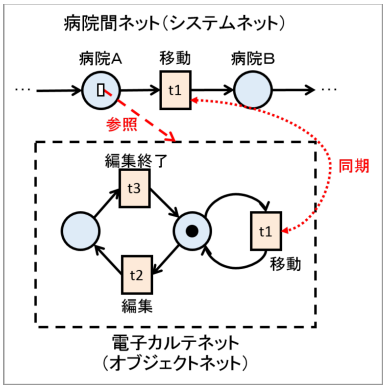

図 2: オブジェクト指向ペトリネットによるワークフロー表現の例

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々の要素の振る舞いを表し，システムネットでそれらの協調の様子を表すことができる．つまり，2 階層以上を持つことができるため，複数の組織にまたがったワークフローやその中を動く主体・情報資源の個々の状態遷移を表すことができる．また，ペトリネットにおいて状態遷移はマーキングとして扱うことができるため，文脈は指定したマーキングを作り出すワークフローのサブネットで表すことができる．

そこで本研究では，トランジションの発火系列を使用して文脈を表現する．特に，サブネットで発火可能な発火系列の集合のことを文脈パターンと呼ぶ．一般的に，情報資源を意味するオブジェクトネットと，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行われる際に情報アクセスが行われる．ここで，発火系列が文脈パターンにマッチする場合は，その発火系列がサブネットで発火可能な発火系列の集合の要素となる場合である．また，文脈パターンとしてのトランジションの発火系列は正規表現を用いて記述し，正規表現によるマッチングを行う．このマッチングにより，文脈に基づいた状況判断を行っている．

3.2. 認可決定モデル

CxAC の認可決定モデルは，XML 関連技術の標準化団体である OASIS（Organization for the Advancement of Structured Information Standards）によって標準化された，情報アクセスに関する制御ポリシを記述するための言語仕様である XACML（eXtensible Access Control Markup Language）の認可決定モデルの一部を拡張して構成している．CxAC では，主体や情報資源や環境の属性情報を扱う PIP(Policy Information Point) の部分を拡張し，CIP(Contextual Inforamation Point) という部分を新たに導入している．3.1. で述べたペトリネットの発火系列からワークフロー情報を取得し，認可決定に利用できるようにしてある．

また，CxAC では XACML のポリシ設計を参考にしている.RBAC では，ユーザ，ロール. セッション，パーミッションが属性として存在し，ユーザにはロールが割り当てられている. 一方 XACML では，ルー

ル，ポリシ，ポリシセットから構成されており，RBAC を XACML で実装するには，ルールやポリシに含まれる target にロールとパーミッションを記述することになる．セッションはアクセスごとに自動的に XACML により生成されるため記述は不要であり，ロールは target の subject 部分に記述する．パーミッションはルールに記述し，対象は target の resource 部分に、命令は target の action 部分に記述する．さらに，CxAC では，XACML のポリシ設計を拡張し，ポリシ上に Condition タグを記述することで文脈を指定できるようにしている．まとめると，ポリシはパーミッション判定，対象，条件 (Condition)，ルールの集合で表されることになり，発火系列から Condition を満たしているかを判断し，かつ例えば「医者」が「カルテ」を「編集する」というルールを満たしている場合にのみ，アクセス許可が出される．このポリシの例を図 3 に示す．この例では，「発火系列が Ambulance」のとき，「主体 rescue_worker」が「資源 consultation か allergiesInfo」に「read か edit」を行う場合，Permit されるということを示す．

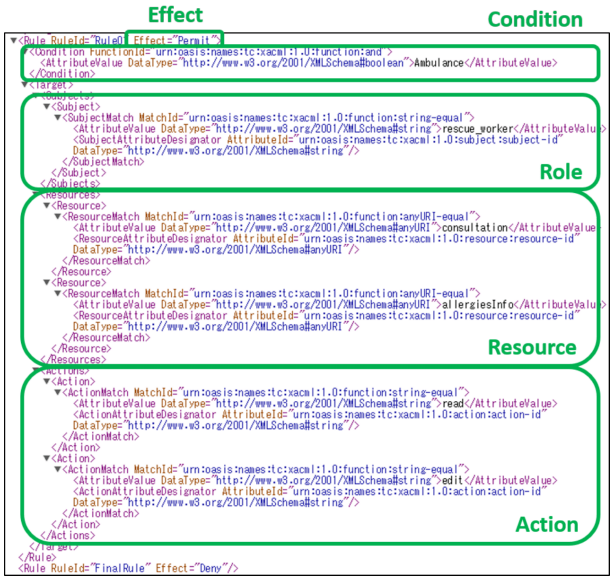

図 3: ポリシに記述されているルールの例

また，認可決定モデルに与えるアクセス要求は，xml 形式で表される図 4 のようなリクエスト構文で記述する．このリクエスト構文には，「主体 doctor が資源 allergiesInfo に対して動作 read を要求している.」といった内容が記述されており，アクセス権限を求めるユーザ側はこの構文を認可決定モデルに送ることになる．

```
<Request>
  <Subject SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>doctor</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
        DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <AttributeValue>allergiesInfo</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
        DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>read</AttributeValue>
    </Attribute>
  </Action>
  <Environment/>
<Request>
```

図 4: リクエスト構文の例

4. 実験・評価

今回，3.1. で示したオブジェクト指向ペトリネットの発火系列を用いて，3.2. で示した XACML を主体や情報資源や操作だけでなく，状況から認可決定を行えるようにしたが，機能を拡張した状態でもアクセスが集中した場合に一定以上の応答性があることが求められる．そこで，同時に多くのアクセス要求を送信した場合に対応できるかについての実験を以下の条件下で行った．

- 「ペトリネット上の 1 つのトランジション『a』を発火させた後，定められたポリシに従ってアクセス要求が正しい権限を持つものかどうかを判断し，許可，不許可の決定を行う」という部分の処理に要する時間を，送信するアクセス要求の個数を変更させながら測定する
- 全てのリクエストの処理時間を平等にするために，発火させるペトリネットは全てのリクエストで毎回生成させている
- アクセス要求の処理を同時に行ったときの時間を測定するため，並列処理の効率化などを行わず，全てのアクセス要求を同時に処理させる
- ポリシについては,「トランジション a が発火している状態で」「test」が「letter」を「read」しようとした場合に「Permit」の判定を出すというルールのみで構成されている，最も単純なものを使用する
- リクエストについては，主体「test」が資源「letter」を動作「read」しようとしているという，ポリシに沿った内容を送信する
- Intel Core m3-6Y30(0.90GHz)　メモリ 4 GB の PC を使用する

今回の実験は図 5 で示すような結果になった. アクセス要求のリクエスト数が 1〜300 までほぼ線形に所要時間が増えていくことが確認できた．リクエスト数が 300 を超えた場合の所要時間はハードウェア側の問題で測定することができなかった．この結果から，外部化されたシステムとしてアクセス制御の運用を行った場合に多くの施設からのアクセス要求のリクエストがあっても，十分な応答性が見込めることがわかった．

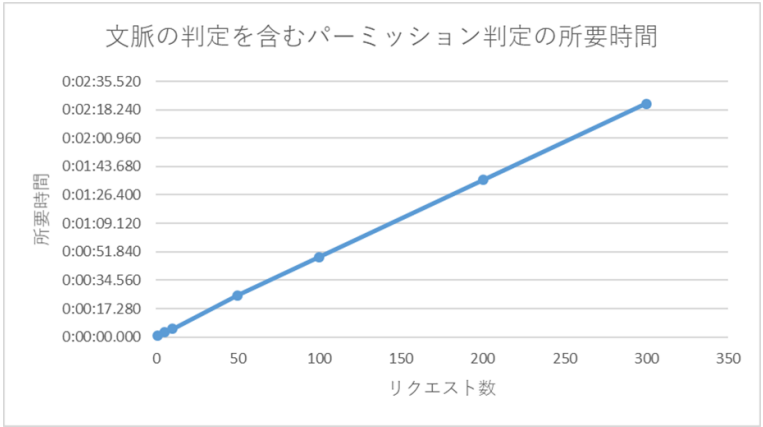

図 5: 文脈の判定を含むパーミッション判定の所要時間

5. 今後の課題

今回の評価実験ではアクセス要求のリクエストやペトリネットの発火が最も単純な場合についての応答時間を測定したが，より複雑なアクセス制御が求められた場合にも大量のアクセス要求のリクエストに対応できる必要がある. そのような実験・評価を行うことと，機能が複雑化してしまっているため，ユーザーにもわかりやすくなるように，複雑化してしまった機能の整備を行うことを今後の研究課題としていきたい．

参考文献

[1] R. S. Sandhu, E. J. Coyne, et al: “Role-based access control models”, IEEE Computer, vol.29, no.2, pp.38-47, 1996.

[2] R´udiger Valk: “Object Petri Nets Using the Nets-within-Nets Paradigm”, LNCS 3098，pp.819-848, Spriner, 2004.