XACML国際標準化とセキュリティー・ポリシー管理

工藤 道治

IBM Research Tokyo

Michiharu Kudo

XACMLは，OASISで策定されたアクセス制御ポリシー言語の国際標準である．2003年に最初の標準が公開されてから，数多くのソフトウェア製品や企業，政府機関においてアクセス制御の標準言語モデルとして採用されてきた．本稿では，XACMLの設計思想と仕様について解説すると共に，OASISにおける国際標準化の経緯について説明する．さらに，XACMLを使ったセキュリティー・ポリシー管理の考え方について述べ，その機能を実現したIBM Tivoli Security Policy Manager製品についても説明する．

XACML is an international standard for access control policy description language standardized in the OASIS standard body. After the XACML version 1 is released in 2003, this standard has been applied in various companies and government organizations as a standard policy description language for IT systems. This paper explains the design principles and technical specifications of the XACML as well as the chronological history of the standardization in the OASIS. In addition, this paper describes the notion of the security policy management using XACML and IBM Tivoli Security Policy Manager as one of the policy management software product.

Key Words & Phrases：XACML，セキュリティー・ポリシー管理，標準化，OASIS
XACML, Security Policy Management, Standardization, OASIS

1. はじめに

XACML（eXtensible Access Control Markup Language）［1］とは，米国の標準化団体であるOASIS（Organization for the Advancement of Structured Information Standards）における技術委員会の1つである．この委員会では，データやIT機器に対するアクセスをどのような条件で誰に許すかというポリシーをXMLで記述するための言語モデルである「XACML」言語の標準化作業を2001年から継続して行っている．OASISは，1993年にグローバルな情報社会のオープン標準を開発，統合および採用を推進する非営利国際コンソーシアムとして設立され，今まで52の技術委員会が標準化作業を完了し，現在でも60以上の技術委員会が積極的に活動を続けている．特に，SOA（Service Oriented Architecture）やWebサービスに関する数多くの業界標準を策定してきたことで知られている．

XACMLが標準化したアクセス制御という技術分野は，1960年代の後半から研究が始まり，その後ISO 7498-2［2］やISO/IEC 10181-3［3］などの国際標準が制定され，それらに準拠した多くのアクセス制御用ソフトウェアが市場に流通している．アクセス制御という考え方は，元は諜報機関や軍の機密を扱う情報システムに必要なセキュリティー要件の1つであり，データに対するアクセスを厳密に管理する必要のある特殊なIT環境を持つ企業・組織以外では重要な技術とは考えられていなかった．しかし，インターネットが普及してから機密情報漏えいやプライバシー情報の不正利用などの情報セキュリティー事件が頻発し，企業のシステム管理者や一般ユーザーまでもが情報セキュリティーを意識しなければならなくなり，今では必要不可欠なセキュリティー技術の1つになっている．

アクセス制御技術はIT機器のさまざまな場所で使われており，24時間365日，常に動作している．例えば，オペレーティング・システムやデータベース，Webサーバーだけではなく，各種ネットワーク機器やファイヤーウォール，グループウェアやソーシャル・ネットワーク・システム，クラウドなど，ありとあらゆる場所でその技術が使われている．このように幅広く使われるようになるにつれて，アプリケーションやIT機器にはベンダー独自のアクセス制御のモデルや仕組みが入っていった．そのため，異なる製品間ではアクセス制御のモデルや用語に統一性や相互運用性がまったくないという状況に陥り，IT統制とい

提出日：2012年3月14日

う観点だけではなく管理コストも増大するという問題が起きてしまった．XACML は，このような「混沌」とした現状を打破するために生み出されたアクセス制御の世界共通語，いわゆる“lingua franca”として機能することを目指して策定された標準である．XACML の仕様には，従来の典型的なアクセス制御にはなかった新しい概念や機能も盛り込まれている．つまり，共通語の役割を果たす「骨格」と，新しい「試み」の 2 つの要素が組み合わされて大きな価値を創り出したと言えるだろう．

筆者は，2001 年の XACML 技術委員会の設立から 2005 年の XACML v2.0 公開までの約 5 年間，本技術の標準化に深くかかわった．本稿では，XACML の標準化のプロセスをたどりながら技術的なポイントについて述べると共に，セキュリティー・ポリシー管理用に開発されたソフトウェア製品についても説明する．XACML の標準化に関する内容は，情報処理学会デジタル・プラクティスにより詳しく掲載されている［4］．

2. アクセス制御ポリシー

2.1 従来技術と課題

アクセス制御ポリシーは，「誰に対して，どの計算機資源にどのような条件の下でアクセスを許可するか（または拒否するか）」を表現する．例えば，「Alice がファイルを読み込む」のようなシステム・イベントが発生したとき，その都度アクセス制御ポリシーを確認して権限の検証を行う．従って，ソフトウェアが扱う対象リソースの種類によって，異なるアクセス制御のモデルでポリシーが記述されるのが普通である．例えばソフトウェアが HTTP サーバーであれば，

```
allow from all, deny from bad.com
```

と記述することで，bad.com のようなネットワーク・ドメインからのアクセスを禁止する．また Java 仮想マシンであれば，

```
Grant Principal com.ibm.security.HWPrincipal “Alice”{
  permission java.util.PropertyPermission “java.home”, “read”
  permission java.io.FilePermission “diary.txt”, “read”
}
```

と記述して，ユーザー“Alice”に java のホームディレクトリや diary.txt ファイルに対する read 権限のアクセスを許可する．このように，個々のソフトウェアやオペレーティング・システムごとにモデルや記述言語が異なるため，企業や組織の中で使われているすべての IT 機器におけるアクセス制御を統一的に管理するためのコスト，「情報セキュリティー統制」にかかる費用や手間が膨れ上がってしまう．もし統一的なモデルや言語ができれば，統制に必要なコストは削減され，ヒューマン・エラーが減り，さらにはソフトウェア・ベンダーが開発してきた類似のアクセス制御機能を統一することで開発コストを大きく削減することも可能になる．メリットが大きいにもかかわらず，2000 年の時点でそのような「ユニバーサル」な標準や技術は存在しなかったのである．

2.2 設計思想

ここからは，標準化の経緯に沿って XACML 標準の仕様と技術の説明を行う．XACML 技術委員会が創設された 2001 年当初，どのような分野で標準が求められているかを調べるために，典型的な適用事例を収集した．ファイル・システムや Web サーバーにおけるアクセス制御ポリシーの記述事例，筆者の研究チームが開発していた XML 文書に対する要素単位のアクセス制御の事例，企業向け Java 言語の 1 つである J2EE（Java 2 Enterprise Edition）におけるセキュリティー・ポリシーの記述事例，XML による医療カルテに対するセキュリティー・ポリシーの事例などが集められ，どのようなアクセス制御モデルを作ればよいか，その設計思想に関する議論が行われた．以下は当時のメンバーが持っていた設計思想に対する共通のビジョンである．

- アクセス制御の基本概念を的確に表現できる．
- 言語のコアセットのみで幅広いアプリケーションのポリシーが記述できる．
- 柔軟なアクセス制御ポリシーの記述を可能とし，かつその解釈にあいまいさを残さない．
- アクセス判定を高速に行う機構を提供する．
- ユーザー固有のアクセス制御判定セマンティクスを記述できる拡張機能を提供する．
- 既存の標準技術をできるだけ再利用する．

上記の設計思想に必要不可欠な部分を XACML 言語の必ず実装する「コア機能」とし，それ以外の機能を「オプション機能」として定義するという考え方がメンバーの中でまとまっていった．

2.3 アクセス制御アーキテクチャー

次に，アクセス制御を行う構成要素のアーキテク

チャーを議論していった．ISO/IEC 10181-3 のような既存の標準を参考にして図 1 に示すモデルを定義した．主要な構成要素は，ポリシー執行ポイント（PEP），ポリシー決定ポイント（PDP），ポリシー管理ポイント（PAP），ポリシー情報ポイント（PIP）である．PEP はアクセス制御の対象となる資源を管理する Web サーバーやデータベースなどの機能要素，PDP はアクセス制御のための判定のみを行うアクセス制御モジュールなどの機能要素，PAP はアクセス制御ポリシーを管理する機能要素，PIP はアクセスを決定する際に使われる参照情報を管理する機能要素である．このアーキテクチャーに特徴的な要素は，図中のコンテキスト・ハンドラーと責務サービスだろう．前者は PDP に渡すアクセス判定要求（図 1 の要求コンテキスト）に使うデータをポリシー情報ポイント（PIP）と協力して組み立てる．後者は PDP のアクセス判定結果（同図返答コンテキスト）に含まれる可能性のある「責務」を強制実行する機能要素である（責務の機能については 2.7 節参照）．このアーキテクチャーの中で XACML 標準が定義しているデータ要素は，XACML ポリシー，要求コンテキスト，および返答コンテキストの 3 つのフローであり，それぞれ W3C で定義された XML Schema を用いてデータ構造のスキーマが定義されている．

2.4 要件と仕様

前節で説明した設計思想とアーキテクチャーに基づき，個々の言語要素と仕様を決めていった．

- アクセス制御規則（XACML 言語で定義された Rule 要素，以下同様）は，アクセス主体（Subject 要素），資源（Resource 要素），動作（Action 要素）の三つ組を中心として記述する．複数のアクセス制御規則をまとめたものがアクセス制御ポリシー（Policy 要素）であり，さらにそれらをまとめたものがアクセス制御ポリシーセット（PolicySet 要素）である．
- アクセス制御規則には任意の条件式（Condition 要素）が記述でき，条件式はあらかじめ用意した関数（Match 要素）を使って記述する．
- すべてのデータは型付きデータとして扱う．
- アクセス制御規則やポリシーの Target 要素のパラメーターを使って検索用インデックスを生成できるようにし，アクセス判定の高速化を可能にする．
- アクセス判定要求は，任意の属性（AttributeId 属性）と属性値（AttributeValue 要素）のペアで構成する．
- アクセス判定結果（Effect 属性）の決定方法は，汎用的なアクセス決定アルゴリズムを用意する（CombiningAlgId 属性）．ユーザー定義のアルゴリズムも同じ枠組みで指定可能とする．

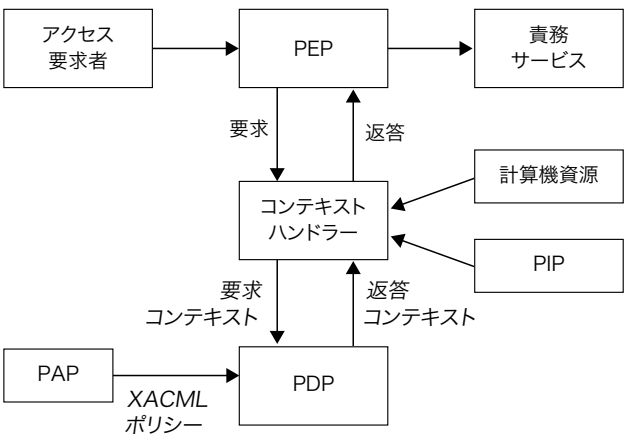

図1. XACMLのデータフロー・モデル

2.5 アクセス制御ポリシーの記述

XACML アクセス制御ポリシーの枠組みについて説明する．Rule 要素は，アクセス制御ポリシーを構成する最小機能単位であり，誰がどの資源に対してどのようなアクセスができるかを，主体，資源，動作の各適用条件を AND，OR，NOT などの論理演算子を用いて AnyOf 要素下に記述する．Rule 要素の Effect 属性には，与えられた要求コンテキストを評価して Rule が返すべきアクセス許可（Permit）かアクセス拒否（Deny）の判定結果を指定する．Policy 要素は，複数の Rule 要素をグループ化する単位であり，特定の資源に対するアクセスを決定する規則をまとめる単位として利用することを想定している．Policy 要素の RuleCombiningAlgId 属性には，複数の Rule から "Permit" のみが戻されるか，"Deny" のみが戻されるか，あるいは両方戻されるかに依存して，最終的に出力すべき結果を計算するアルゴリズムを URI（Uniform Resource Identifier）を使って指定する．PolicySet 要素は，PolicySet 要素あるいは Policy 要素をまとめるための機能単位であり，構造は Policy 要素とほぼ同一である．Target 要素は，Rule 要素，Policy 要素，PolicySet 要素の中に記述し，要求コンテキストに対して，どの Rule や Policy が適用可能かを判定するために用いる．XACML アクセス制御ポリシーの枠組みを図 2 に示す．なお，紙面の関係で XML 要素の閉じタグや属性値などは一部省略している．

```
<Policy RuleCombiningAlgId = “deny-overrides”>
 <Target>
  <AnyOf>
   <AllOf>
    <Match>...
    <Match>...
 <Rule RuleId = “R1” Effect = “Permit”>
  <Target>
  <AnyOf>
    <AllOf>
      <Match>...
      <Match>...
  <Condition>
 <Rule RuleId = “R2” Effect = “Deny”>
  ...
 <ObligationExpressions>
  ...
</Policy>
```

図2. アクセス制御ポリシーの枠組み

2.6 ルール結合アルゴリズム

RuleCombiningAlgId 属性は，複数のルールから出力されるアクセス判定結果をどのように結合して最終的な返答とするのか，アルゴリズムのタイプを指定する．"Deny-overrides" は，"Permit" と "Deny" の両者が戻された場合，"Deny" を優先するアルゴリズムであり，"Permit-overrides" はその逆である．XACML v1.0 では 4 つのアルゴリズムを定義していたが，最新の XACML v3.0 では 12 個のアルゴリズムに拡張されている．アルゴリズムのタイプはユーザーが自由に拡張でき，それが XACML の機能的な特徴の 1 つにもなっている．拡張するには，ドメイン特有なロジックに対応する URI を定義し，そのアルゴリズムの解釈エンジンを実装すればよい．

2.7 責務

責務（Obligation）とは，特定の資源に対するアクセス要求に伴って必ず実行が必要な処理を記述する概念である．これは，筆者らが提案した必須処理付き認可モデル（Provisional Authorization Model）における必須処理（Provisional Action）と呼ばれる概念と同一である［5］．必須処理付き認可モデルでは，従来の「誰にどのような処理を許可する」といった記述より，もっと柔軟に規則を記述できる．例えば「社外ユーザーはファイル A を更新できるが，更新時には必ず暗号化処理を行いかつアクセスログを残す必要がある」というようなセキュリティー規則も記述できるようになる．社外ユーザーである Alice がファイルを更新しようとした場合，「更新は許可，ただし暗号化とアクセスログ記録が責務」というアクセス判定結果が出力され，PDP から PEP に戻される．「ただし」以降の責務の記述が PEP によって責務サービスに送られ，データの暗号化やアクセスログ記録などの処理が行われる．PEP と責務サービスは同じマシン上に存在してもよい．このように責務をアクセス制御モデルに組み合わせると，通常のアクセス制御という目的以外にも，データの完全性や機密性保持の仕組みの実現，デジタル著作権管理のための実装，プライバシー・ポリシーの実装，アクセスに対する課金メカニズムとの連携など，幅広い Web アプリケーションの動作をアクセス制御と関連させて記述することが可能になる．

3. 適用事例

ここでは，医療分野で XACML を使った適用事例について紹介する．医療カルテには患者のプライバシー情報が非常に多く含まれており，高度な機密事項を扱う必要がある．カルテにアクセスするユーザーは，医師や看護士以外にも患者や病院の関係者，政府当局など多岐にわたり，患者の容態に応じてアクセス制御のポリシーも大きく変化する．例えば容態が安定している場合はカルテに対して担当医のみが読み書きできればよいが，容態が急変すれば担当医以外の複数の医師もカルテの全情報に遅滞なくアクセスできなければならない．医療の現場における複雑で高度なセキュリティー要件を満たすことができれば，XACML の高い適用性を証明することができる．

そこで，XACML 技術委員会と米国健康情報技術標準化委員会（Health Information Technologies Standards Panel）が協力し，2008 年の RSA コンファレンスにおいて XACML v2.0 を用いたアクセス制御機能を中心にしたデモンストレーションを行った［6］．米国には病院などの医療情報を扱うシステムのために HL7（Health Level 7）と呼ばれる標準があり，その中には医療カルテのアクセス制御に関するさまざまなポリシーが要件として記述されている．このデモでは，ロールに基づくアクセス制御（Role Based Access Control），医療カルテの部分的なデータ開示，プライバシー保護，コンセント・コードのサポート，緊急事態による変更対応，機密データのフィルター処理など各種のアクセス制御の要件を XACML v2.0 の仕様を用いて記述し，実行してみせた．さらに複数の医療機関の間でポリシーを交換し，異なる実装上でも同じ動作をするという「相互運用性」のデモも行った．Web サービスの標準や SAML（Security Assertion Markup Language）といったメ

ジャーな標準技術と一体となったこのデモの成功により，XACML が目指した「アクセス制御の共通言語を作る」という目標がある意味で達成できたと言えるだろう．

4. 国際標準化

4.1 XACML 誕生まで

World Wide Web コンソーシアム（W3C）が XML に関する最初の仕様を勧告した 1998 年当時，IBM の東京基礎研究所では XML に関する「セキュリティー」が次の技術課題になると考え，XML 文書のデジタル署名（XML Signature）や暗号化（XML Encryption）に必要なセキュリティー技術の研究と標準化を進めていた．当時筆者の研究チームでは，XML 文書に対するアクセス制御技術の研究を進めており，XML アクセス制御言語（XACL）という言語を開発して外部に公開していた［7］．XML 文書は，データを構成する個々の要素に対してタグ付けを行って，データ構造の可読性を向上させることができる．XACL は，そのタグ付け機能を利用して文書の内部構造に対する粒度の細かいアクセス制御を実現しようとしたものである．XACL 言語仕様をネットに公開し，論文の査読管理アプリや医療カルテの開示制御などを適用例として提示した．XACL の新しい試みの 1 つは「必須処理付きアクセス制御」を実現していることである．この概念は，前述のように責務として XACML の仕様に取り込まれた．

2000 年当時，XML に関するアクセス制御技術の研究は海外の大学でも行われており，筆者は積極的にそれらの大学と研究交流を行っていた．イタリア・ミラノ大学の Ernesto Damiani 教授の研究グループとは，技術的な内容だけでなく，互いの研究成果をいかに世の中に出し，普及させていくことができるかといった議論も行い，そのような雰囲気の中でこの技術を国際標準にできないかと考えるようになっていった．XML に関する標準は W3C が中心になって行っており，当時は XML 署名と暗号化が XML のセキュリティーに関する二大標準化トピックであった．2 つの注目度があまりにも高かったため，アクセス制御のトピックを W3C で始めても，興味を持つ人が集まらないかもしれないという危惧を持つほどであった．そこで ebXML（Electronic Business using eXtensible Markup Language）や SAML といった XML を使った Web サービスに関する標準化を積極的に行っていた標準化団体である OASIS に注目し，アクセス制御に関する標準化の母体とすることにした．そして，Damiani 教授らと共に 2001 年 4 月に XACML 技術委員会を OASIS に設立したのである．なお XACML の呼び方だが，「エクザクムル」（“ザ”にアクセント）と発音する．

4.2 XACML 技術委員会における標準化

OASIS の標準化プロセスは，1）技術委員会の設立と設立趣意書の定義，2）技術仕様の十分な議論と委員会ドラフトの策定，3）パブリック・コメントの収集と対応，4）OASIS 標準化構成員による標準化への最終投票，5）OASIS 標準として公表，という流れである．XACML 設立当初は，議論に毎回参加する積極的な参加者はおおむね 7-8 名であり，そのうち企業からの参加者は筆者を含め 4-5 名，残りは大学関係者だった．当時，筆者は自分の研究の一部として標準化を行っており，しがらみのほとんどない「理想的な」状態だったといえる．自分が正しいと信じることを主張し，親しくしていた大学関係者と一緒になって反対意見に反論するというようなスタイルであった．委員の間で意見が大きく 2 つに分かれた仕様の例としては，Rule や Target 要素下に記述する「主体」，「資源」，「動作」というアクセス制御基本三要素を言語要素として明示的に保持すべきか，あるいは汎用的な仕組の中で暗黙的に記述すべきか，という対立があった．数ヵ月に渡る長い議論の後，この標準がアクセス制御専用のものであるのだから，基本三要素は明示的に記述すべきという結論に至った．当時は適切な判断だと思ったのだが，XACML v3.0 では三要素は明示的には区別しなくなった．時代の要請によって仕様は変わっていくことを示した一例であろう．このように 1 つの仕様を巡って数ヵ月を要することもある．2003 年と 2005 年にそれぞれ OASIS 標準になった XACML v1.0 と v2.0 を仕上げるためにかかった期間は両方とも約 2 年であり比較的短期間であった．XACML v3.0 はまだ最終的な OASIS 標準にはなっていないが，議論に 6 年以上かかっている理由としては，より多くの企業・組織が標準化に参入したこと，仕様の規模が大きくなりかつ複雑度も増したこと，などが挙げられるだろう．

4.3 産業界・学会へのインパクト

現在，XACML の技術委員会には，IBM, Microsoft, EMC, Oracle, Cisco, Redhad といった IT ベンダーが参加しており，公開される技術仕様は実質的な業界標準になっている．2007 年当時，60 以上の企業・組織が XACML を利用することを正式に表明していたが，年を経るごとにサポートする企業・組織の数は増えていっ

た．産業界に対しては，当初の予想をはるかに上回る大きなインパクトを出すことができたと思う．

大学や学会などのアカデミアに対してはどうだろうか．Google Scholar で“XACML”を検索すると，7,000を超える論文やプレゼンテーションがヒットする．“XML Signature”や“XML Encryption”の検索結果に対して約二倍の数がヒットすることを考えると，アカデミアに対してもそれなりのインパクトを与えることができたと言えるだろう．また国際学会の査読を引き受けると，XACMLに影響を受けて始まった研究や論文が数多く投稿されており，今までに数十もの関連論文を査読した．これは国際標準がきっかけとなり，大学などのアカデミアにも影響を与えた典型的な例である

5. セキュリティー・ポリシー管理

企業や政府機関のような大きな組織では，多種多様なIT 機器・ソフトウェア製品が使われている．XACMLのような標準共通言語を利用すれば，組織全体で統一してアクセス制御ポリシーを管理・設計することができるようになる．ここでは，Web アプリケーションを例にとって，統合されたセキュリティー・ポリシー管理の利点について説明する．

図 3 は，典型的な Web アプリケーションの構成例を表している．クライアント PC から Web アプリケーションへのアクセスが発生すると，サーバー側では，プロキシやポータル・サーバーでのユーザー認証やアクセス認可を経て，アプリケーション本体やバックエンドの基幹システムで処理が実行される．この際，プロキシやポータル，既存アプリケーション内では，それぞれの製品の仕様によってユーザーの認証方式やアクセス制御の仕組み・ポリシーの記述方法がばらばらであることが多い．これらに対して，統合セキュリティー・ポリシー管理機能を導入することで，アクセス制御ポリシー記述の整理・統合・置き換え，さまざまなポリシー決定ポイント（PDP）との連携，見通しのよい IT 機器のポリシー変更管理などが可能になる．それらの統合の結果として，より成熟度の高い IT 統制を実現することができる．

昨今では，このような統合されたセキュリティー・ポリシー管理のためのソフトウェア製品も出てきている．例えば，IBM 社の Tivoli Security Policy Manager（TSPM），オラクル社の Oracle Entitlement Server（OES），エントラスト社の Entrust GetAccess などが挙げられる．

TSPM は IBM Web Application Server（WAS）と組み合わされ，Enterprise Java アプリケーションとして実行される．ここで TSPM の代表的な 3 つの機能要素について説明する．一番目は，複数の IT 機器で使われるセキュリティー・ポリシーを統合して管理するための PAP（ポリシー管理ポイント）としての機能要素である．二番目は，セキュリティー・ポリシーに基づき実行時にアクセス決定を行う PDP（ポリシー決定ポイント）としての機能要素である．三番目は，実行時に処理を行うアプリケーションの中でアクセスを執行する PEP（ポリシー執行ポイント）としての機能要素である．この場合，J2EE 1.4 以降に含まれる Java Authorization Contract for Containers（JACC）と呼ばれる Java プラグイン機能や，Java API for XML Based RPC（JAX-RPC）と呼ばれる遠隔プログラム呼び出しを行う API を使って PDP をアプリケーションの中からシームレスに呼び出して利用する．TSPM 製品については，開発者による詳細

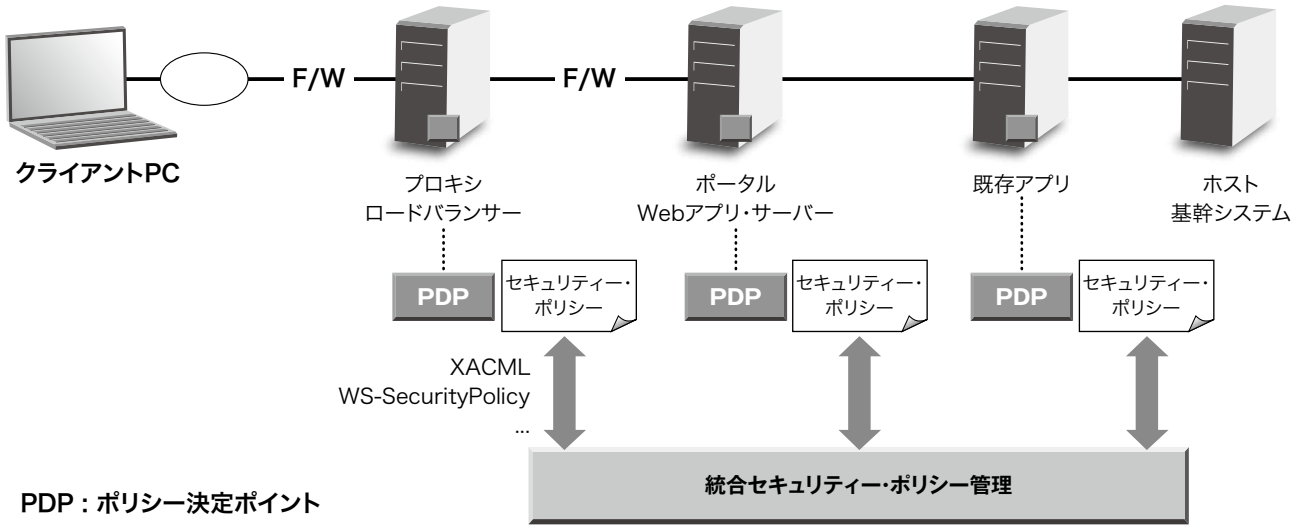

図3. Webアプリケーションへのセキュリティー・ポリシー管理の適用例

な技術説明書が公開されているので，そちらも合わせて参照されたい［8］.

XACML のようなアクセス制御ポリシーの共通言語ができる前は，統合的なセキュリティー・ポリシー管理の実現は容易ではなく，それが IT 管理コスト増大の一因になっていたと推測できる．しかしこれからは，システムが XACML の仕様に準拠していれば，どのようなソフトウェアやサービスであっても，誰が，どのリソースに，どのような条件のものでアクセスできるか・利用できるかを，非常に見通しよく管理することができるようになるだろう.

6. おわりに

本稿では，筆者が深くかかわった XACML という標準について，技術仕様と標準化の経緯，および関連ソフトウェア製品について述べた．最新の XACML v3.0 の仕様は，技術委員会のドラフトが 2010 年 8 月に完成したが，現在も活発に議論が続いている．XACML は仕様が複雑すぎる，機能的に不足しているなど，まだまだ課題が多く残っていることも承知しているが，アクセス制御の共通語を作るという当初の夢は，長い時間はかかったがある程度実現できたと思う．このような標準によって混沌とした IT の世界を少しでも「整える」ことに貢献できたとしたら大変うれしいことである．その中で，2011 年の European Identity コンファレンスにおいて XACML v3.0 の活動が“最も大きな影響を与えた標準化賞（Influential Standardization Efforts Award）”を受賞した．XACML の活動によって多くの人や組織がアクセス制御の技術分野に注目するようになったとすれば，一人の研究者として大きな喜びを感じる.

参考文献

［1］ OASIS eXtensible Access Control Markup Language（XACML）TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml/

［2］ ISO 7498-2:1989, “Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture”（1989）.

［3］ ISO/IEC 10181-3, “Information technology – Open Systems Interconnection – Security frameworks for open systems: Access Control Framework”, International Standard（1996）.

［4］ 工藤道治,”XACML アクセス制御ポリシー言語の国際標準化”, 情報処理学会デジタル・プラクティス，Vol.2, No.4, Oct, 2011.

［5］ Michiharu Kudo, Satoshi Hada, “XML document security based on provisional authorization”, 7th ACM conference on Computer and communications security（2000）.

［6］ OASIS, “OASIS 会員が HITPS 健康管理のシナリオで XACML アクセス制御標準の相互運用性をデモンストレーション”, http://www.oasis-open.org/jp/news/oasis_news_04_07_08.php

［7］ XML Access Control, http://www.trl.ibm.com/projects/xml/xss4j/docs/xacl-readme.html（2000）.

［8］ Axel Buecker, et. al “IT Security Policy Management Usage Patterns Using IBM Tivoli Security Policy Manager”, http://www.redbooks.ibm.com/abstracts/sg247880.html?Open

日本アイ・ビー・エム株式会社
東京基礎研究所
シニア・リサーチャー

工藤 道治 Michiharu Kudo

［プロフィール］
1988 年，日本 IBM 入社．以来，同社東京基礎研究所にて，情報セキュリティー・ポリシー,トラステッド・コンピューティング,セキュリティー・マネジメントに関する研究や国際標準化などに従事．OASIS XACML 技術委員会創設メンバーの一人．工学博士,情報処理学会，日本セキュリティー・マネジメント学会，各会員.