社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－SE－151（20）
2006／3／24

モデル駆動型開発に基づくセキュリティ設定 － Web サービスのための認証設定 －

佐藤 史子, 中村 祐一, 小野 康一

日本 IBM 東京基礎研究所
{sfumiko, nakamuray, onono}@jp.ibm.com

サービス指向アーキテクチャ（SOA）では，サービスを組み合わせることにより新たなアプリケーションを開発する．我々は，SOA のセキュリティ基盤として使われる WS-Security を設定するフレームワークとして，モデル駆動型セキュリティ（MDS) を提案している．MDS では，プラットフォームに依存しないアプリケーションモデル上でセキュリティを設定し，モデル変換によりセキュリティポリシーを生成する．これまでの研究では署名・暗号化によるメッセージ保護を対象にしていたが，本研究では認証設定に MDS を適用した．WS-Security の認証モデルでは，シングル・サイン・オン (SSO) や ID 伝播が可能であるが，実際に設定を行うには複数のセキュリティドメインの統合や ID を伝播するサーバ間の信頼関係などを考慮する必要があり複雑である．アプリケーションモデル上の認証設定を表す注釈と，複雑な認証プラットフォームの情報をセキュリティ基盤モデルとして定義したことで，認証設定に関してもモデル変換を可能にした．本研究により，認証プラットフォームの詳細を意識せず，認証設定を行うことが可能になる．

Security Configuration Based On Model Driven Architecture - Authentication Configuration for Web Services -

Fumiko Satoh, Yuichi Nakamura, Koichi Ono

IBM Research, Tokyo Research Laboratory

Service-Oriented Architecture (SOA) makes a new application by combining existing services. We proposed Model Driven Security as a framework to configure WS-Security, which is used a security infrastructure of SOA. In MDS, security can be configured in a platform-independent application model, and the security policy is created by the model transformation. We focused on configuring message protection by signature and encryption. In this study, we apply MDS to authentication configuration. The authentication mechanism of Single Sign On or ID propagation can be supported by WS-Security, however the configuration is very difficult in practice. Because it is necessary to consider the federation of multiple security domain and the trust relationship between services of ID propagation. We propose a Qualifier to specify an authentication configuration in an application model, and a Security Infrastructure Model to model complex authentication platform information. This idea becomes a model transformation possible to configure authentication. Our contribution is that the authentication configuration becomes easier without the knowledge of an authentication platform.

1. はじめに

サービス指向アーキテクチャ（Service-Oriented Architecture: SOA）は，サービスを組み合わせることによりアプリケーションを開発するという考え方であり, エンタープライズ・システム開発の基盤として定着しつつある．サービスとは，インターフェースとアクセス方法によって規定されるものであり，WSDL などによって記述される．サービスそのものはその実装手段とは独立に定義されるので, 柔軟なアプリケーション開発が可能になる．

エンタープライズ・システムの開発においてセキュリティの確保は重要な問題であるが，SOA に基づく開発では，これまでとは違う新たな要件を考慮することが必要になってくる．例えば, 複数のサービスの組み合わせを考えると, 各サービスが異なるセキュリティドメイン上で実行されるので，ドメインの統合（Federation）は重要な課題となってくる．

Web サービスセキュリティ (WS-Security[1]) のロードマップドキュメントでは, セキュリティドメイン統合のためのモデルが提案されている．これまでにも，同一の機構に基づく統合は Kerberos などでも提案されているが，異なる機構上のドメインを統合する点がこのモデ

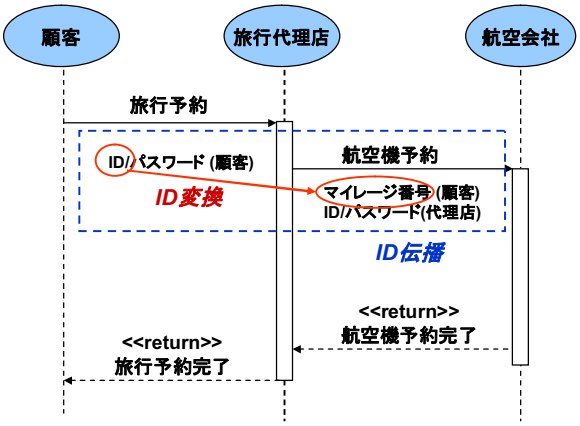

図 1. 旅行予約シナリオ

ルの特徴である．SOA で対象とするエンタープライズ・システムでは，様々なセキュリティ機構が混在するので，この考え方は極めて重要である．

WS-Security により，SOA のセキュリティ基盤は提供できるが，実際にセキュリティを設定することはユーザーにとっては難しい場合が多い．WS-Security のメッセージセキュリティはセキュリティトークンによる署名・暗号化によって実現されるが，セキュリティトークンは，Kerberos や SSL などさまざまなセキュリティ機構を利用して作ることができるよう，非常に柔軟に設計されているため，設定が複雑になるからである．

WS-Security の認証機構としては，SSO や ID 伝播を実現することが可能である．しかし，SSO のために伝播されるユーザーの ID を含むセキュリティトークンと他のセキュリティトークンを区別する方法や，伝播されたユーザーの ID を信頼する方法，異なるセキュリティドメイン間での ID 変換などを考慮した上で，認証設定を行う必要がある．そのためには，プラットフォームのセキュリティ機構や ID 変換のためのサーバの設定など，認証プラットフォームをよく理解しておかなければならない．

一般的なセキュリティエンジニアリングでは，アプリケーションを実装した後にセキュリティポリシーが作成されていることが多く，問題となっている．そこで，アプリケーションのデザイン時からセキュリティを設定する方法として，モデル駆動型開発（Model Driven Architecture: MDA）[2]を元にしたモデル駆動型セキュリティ（Model Driven Security: MDS）を提案した[6]． MDS は，アプリケーションのモデルを設計する段階で，必要なセキュリティを設定し，アプリケーションのモデルからセキュリティポリシーを得るためのフレームワークである．

今までは，MDS では署名・暗号化によるメッセージ保護だけを考慮していた．本研究では，MDS によって認証設定を行う方法を提案する．本研究で新しく導入するのは，以下の 2 点である．

- セキュリティドメインごとに異なる ID をアプリケーションモデル上で統一的に扱うための抽象的な注釈の導入
- ID 変換など認証用プラットフォームに特有の概念を考慮してモデル化したセキュリティ基盤モデル（Security Infrastructure Model: SIM）の定義

本研究により，認証設定も MDS フレームワークを適用して行うことが可能になる．認証を必要とするユーザーの種類を注釈で指定し，認証プラットフォームをモデル化したセキュリティ基盤モデル（SIM）を使ったモデル変換により，セキュリティポリシーが生成できる．Kerberos や ID 伝播，SSO など認証プラットフォーム全体を理解していないとできなかったセキュリティ設定が，プラットフォームを意識することなく可能になることは大きな貢献である．

2 章で認証設定における問題について説明し，3 章で MDS による認証設定の方法，4 章で具体例，5 章で関連研究を示す．

2. 認証設定における問題

ここでは，認証設定を考えるためのサンプルアプリケーションとして，図 1 のような旅行予約シナリオを用いる．

このシナリオでは，旅行代理店，航空会社の 2 つのサービスが連携して ID を伝播する．顧客が旅行代理店に対し旅行予約リクエストを送信すると，旅行代理店は航空会社に航空機予約リクエストを行う．旅行代理店は顧客から送られるセキュリティトークンに含まれる ID/パスワードにより顧客を認証し，航空会社への航空機予約リクエスト時に顧客のマイレージ番号を伝播する．顧客はすでに旅行代理店で認証されているので，ここでは顧客のマイレージ番号のみが航空会社に送られる．このように旅行代理店が顧客を認証し，航空会社は旅行代理店が伝播した ID すなわちマイレージ番号により顧客を確認する．本研究では，この認証方法を

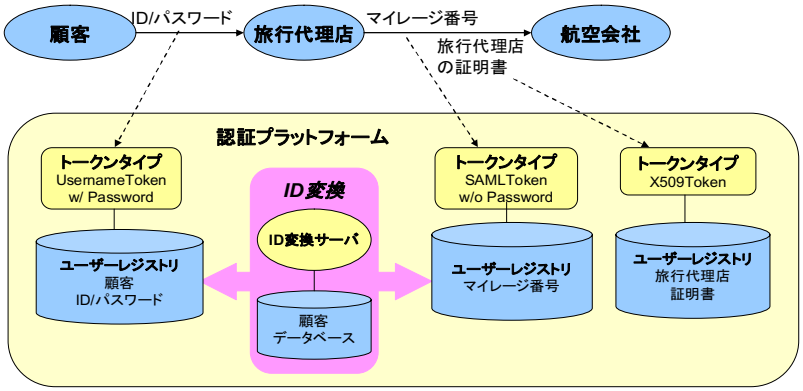

図 2. 認証プラットフォーム

ID 伝播と呼ぶ.

ID 伝播では，旅行代理店は顧客から受け取った ID を航空会社に伝播する際，顧客の ID を航空会社のドメインに対応した ID に変換する必要がある．本研究では，顧客の ID をマイレージ番号に変換することを ID 変換と呼ぶ.

しかし, 航空会社では伝播されてきた顧客の ID が正しいかどうかを判断することができない．そこで，航空会社は ID を伝播する旅行代理店を信頼することによって，ID が正しいと判断する．そこで，航空会社は旅行代理店から顧客の ID を受け取るとともに旅行代理店の認証も行い，旅行代理店の ID/パスワードや証明書など，旅行代理店の認証情報を要求する．本研究では, 航空会社が旅行代理店を認証するときに使う方法を信頼方法と呼ぶ．信頼方法が Basic の場合には，旅行代理店の認証情報として ID/パスワードを要求する．信頼方法が Signature の場合は，旅行代理店の署名を要求することを示す.

ID 伝播を行うための認証プラットフォームを図 2 に示す．旅行代理店は顧客の ID を自身のセキュリティドメインのユーザーレジストリを使って認証する．その後，航空会社に ID を伝播するために, 航空会社のセキュリティドメインでの ID すなわちマイレージ番号に変換する必要があるため, ID 変換サーバへ ID 変換を依頼する．ID 変換サーバは，旅行代理店から受け取った ID を航空会社のマイレージ番号に変換して返す. 旅行代理店はマイレージ番号を航空会社に伝播し, 航空会社は自身のセキュリティドメインのユーザーレジストリで顧客のマイレージ番号を確認する.

旅行代理店，航空会社の各サービスのセキュリティポリシーを設定し, アプリケーション全体の認証設定を正しく行うには, 認証プラットフォームが提供している ID 伝播や SSO などによる認証方法, 旅行代理店と航空会社のセキュリティドメインの違いによる ID 変換の設定, 旅行代理店と航空会社間の信頼方法, プラットフォームがサポートしているトークンの種類など, 認証プラットフォーム全体を把握しておく必要がある．そうでなければ，各サービス間で整合性のとれたセキュリティポリシーを生成することができない.

現状では, これらの設定をほぼ手作業で記述する必要がある．したがって，アプリケーション開発者がセキュリティ設定をアプリケーションの開発・実装段階で行うのは，非常に労力が大きい．以上のことから，解決すべき問題点は以下の 2 点であると考える.

1. アプリケーション開発者が, プラットフォームが提供する認証方法（ID 伝播や SSO など）を理解するのは難しい
2. セキュリティを設定するには，1.の認証を実現する複雑な認証プラットフォームを詳細まで理解しなければならない

3 章では MDS を認証設定に適用し，以上の問題を解決する方法を示す.

3. モデル駆動型セキュリティ

本章では, 3.1 節で認証設定での問題を MDS で解決する方法を述べる．次に，本研究の中心である認証設定のためのセキュリティ基盤モデルの定義と, セキュリティポリシーを生成するためのモデル変換について 3.2 節と 3.3 節で説明する.

3.1 モデル駆動型セキュリティ概要

モデル駆動型セキュリティは, セキュリティ設定をモデル駆動型開発（MDA）で行う手法である．MDA では，プラットフォーム非依存

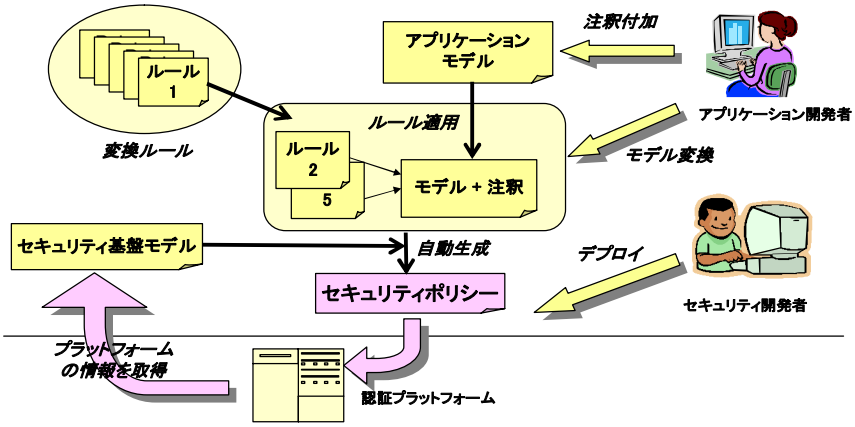

図 3. モデル駆動型セキュリティ アーキテクチャ

のモデル(Platform Independent Model: PIM)とプラットフォーム依存のモデル(Platform Specific Model: PSM)を定義し，PIM からモデル変換により，より詳細な PIM や PSM すなわち実行環境に依存した設定を得ることができる.

2 章で示した認証設定での問題点を解決するために，本研究では図 3 のような MDS アーキテクチャを認証設定に適用する. PIM であるアプリケーションモデル上に必要なセキュリティを表す注釈をつけることで認証設定を行い, 変換ルールに基づくモデル変換によってセキュリティポリシーが生成できる.

本研究では複雑な認証プラットフォームを理解しないとセキュリティ設定ができない, という問題を解決するため, プラットフォームに依存しない表現で抽象的に認証を設定するための注釈を導入した. また，認証プラットフォーム全体をセキュリティ基盤モデル（SIM）としてモデル化した. 抽象的な注釈からセキュリティポリシーを生成するために, モデル変換時に SIM を参照することで必要な認証プラットフォームの情報を得ることができる. したがって, アプリケーション開発者は注釈をつけるだけでセキュリティ設定ができ, 認証プラットフォームの詳細を理解しなければならないという問題を解決できる.

図 4 は認証設定を表す抽象的な注釈の例を示す. この注釈ではサービスを利用するユーザーの種類が traveler である場合，認証が必要であることを示している．このように，アプリケーションモデルに付加する注釈は, 認証プラットフォームに依存したトークンの種類やドメインなどの情報は現れない. 認証設定をプラットフォームに依存しない抽象的な注釈で表現

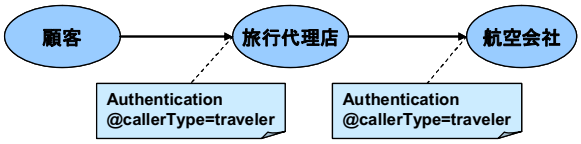

図 4. 認証を表す注釈

することで, 注釈は複数のアプリケーションで使用することが可能になる．例えば，複数のエンタープライズアプリケーションで，callerType=employee のように認証設定を統一できる．これにより，各アプリケーションが利用できる認証プラットフォームが異なっていても同じ認証設定を行うことができ, アプリケーション開発者はプラットフォームの違いを意識する必要はない．したがって，認証プラットフォームが変わっても注釈による認証設定には影響せず, プラットフォームの変更に柔軟に対応することが可能になる.

MDS フレームワークでアプリケーションのセキュリティポリシーを作成する手順は次のようになる.

1. アプリケーションモデルのサービスインターフェースに，注釈を付加する
2. 注釈で表された認証設定が, そのプラットフォームで実現できるかどうかをセキュリティ基盤モデル（SIM）と比較し整合性を確認する
3. アプリケーションモデルからのモデル変換により注釈と SIM から認証用セキュリティポリシーを生成する

3.2 節でセキュリティ基盤モデル（SIM）, 3.3 節でモデル変換について示す.

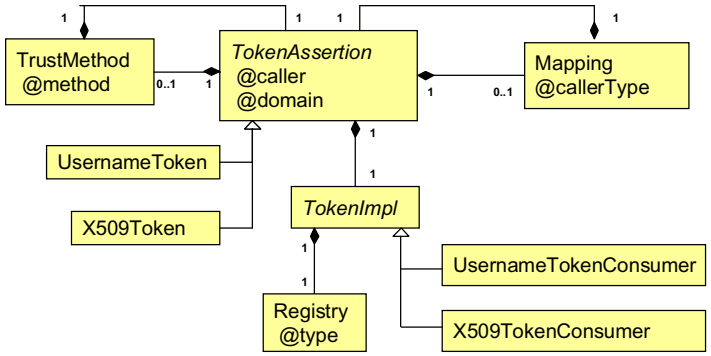

図 5. セキュリティ基盤モデル

3.2 セキュリティ基盤モデル

セキュリティポリシーを生成するには, プラットフォームが提供しているセキュリティ機構を知る必要がある. そうでないと, アプリケーションがデプロイされるプラットフォームで有効なセキュリティポリシーや設定を作ることができない. 特に, 認証設定を考える場合, 異なるセキュリティドメイン間の ID 変換が必要であり, さらに要求されるトークンの種類やドメインなどを理解し, 旅行代理店と航空会社間で矛盾のないセキュリティポリシーを生成する必要がある.

図 2 は認証用セキュリティポリシーを生成するのに必要なプラットフォームの情報を表している. WS-Security でユーザーを認証するためには, ユーザーから送られたセキュリティトークンを処理し, ID を取得して, ユーザーレジストリに問い合わせる必要がある. ここで, ユーザー認証のために要求するトークンの種類や利用するレジストリはアプリケーションがデプロイされるプラットフォームに依存して決定される. したがって, トークンの種類やレジストリなどの認証方法を SIM にモデル化する必要がある. そのほか SIM には, ID 変換のための情報と航空会社が旅行代理店を認証する方法 (信頼方法) をモデル化する必要がある.

本研究では, 図 2 をモデル化する SIM の構造を図 5 のように定義した. 主な要素は次のように定義される. その他の要素はセキュリティ設定ファイルを作成する場合に参照されるもので, 本稿では省略する.

- TokenAssertion 要素 : トークンの種類を指定する. TokenAssertion 要素は WS-SecurityPolicy [3]で定義されている抽象的な要素であり, 実際には要求されるトークンのインスタンスとなる
- Mapping 要素 : 親要素のトークンは ID 変換を必要とすることを示す
- TrustMethod 要素 : 伝播される Caller の ID を信頼する方法を指定する. method 属性により Signature, Basic などのキーワードで指定する

SIM は認証設定に必要なプラットフォーム情報をすべてモデル化したものである. アプリケーション開発者が設定するのは, アプリケーションにアクセス可能なユーザーの種類だけであり, アプリケーションの内容のみに依存している. アプリケーションがデプロイされているプラットフォームが提供している認証機能を考慮せずに認証設定が行え, モデル変換を行えばセキュリティポリシーが生成できる. これは認証プラットフォーム全体をすべて SIM としてモデル化したことで可能になったことであり, これは本研究の貢献のひとつである.

3.3 モデル変換

ここでは, アプリケーションモデルにつけられた注釈から, SIM を参照し認証用セキュリティポリシーを生成するモデル変換について述べる.

本研究の注釈では, アプリケーション開発者が callerType=traveler などのようにアクセスできるユーザーの種類を指定する. ここで, 注釈の callerType は SIM が作成される際に定義されるものとする. アプリケーション開発者は定義された callerType の中から, 認証を行いたいユーザーを選択し, 注釈として設定する. 各 callerType とプラットフォームがサポートしているセキュリティドメインとの対応は, 対応表があらかじめ作られているものとする. この対

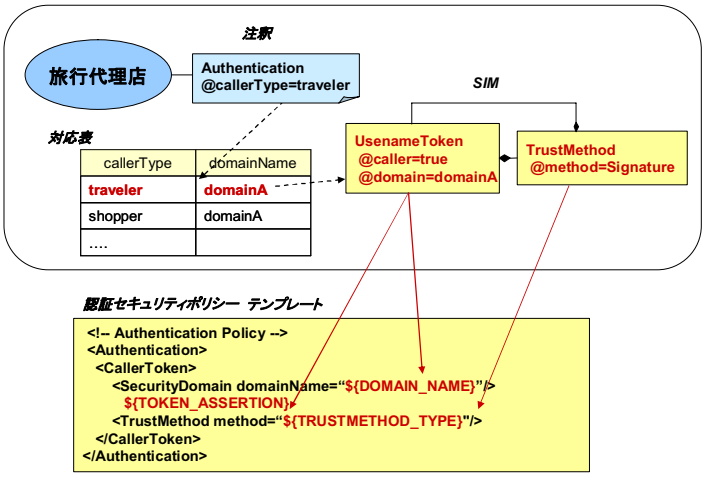

図 6. モデル変換

応表と SIM を用いて，callerType からセキュリティポリシーを生成する．

モデル変換で生成するセキュリティポリシーは，WS-SecurityPolicy [3]を拡張して定義した．標準の WS-SecurityPolicy では，認証のためのセキュリティポリシーが正確に記述できないためである．この認証用セキュリティポリシーの生成には，セキュリティポリシーのテンプレートを使う．テンプレートは，認証用セキュリティポリシーで SIM や注釈に依存する部分を変数にしたものである．この変数を，SIM から取得した値で埋めていくことにより，認証用セキュリティポリシーが生成できる．

これらを利用して，モデル変換の詳細手順は次のようになる（図 6）．

1. 注釈の callerType の値を確認する
2. あらかじめ準備されている対応表から，callerType の値に対応するドメインの値を得る
3. SIM の TokenAssertion 要素のうち，caller 属性が true である要素を抜き出す
4. 3 で抜き出した TokenAssertion 要素のうち，ドメインが 2 の値と等しいものを選ぶ
5. 4.で選んだ TokenAssertion の値をセキュリティポリシーテンプレートに埋める．ここでは${DOMAIN_NAME}にドメインの値，${TOKEN_ASSERTION}に TokenAssertion 要素を埋めることができる
6. 4.で選んだ SIM の TokenAssertion 要素に TrustMethod 要素が関連付けられている場合，セキュリティポリシーテンプレートの${TRUSTMETHOD_TYPE}を埋めることができる．関連付いていない場合，テンプレートの TrustMethod 要素を削除する

SIM の Mapping 要素は ID 伝播が旅行代理店と航空会社の間で可能かどうかを確認するために使われる．旅行代理店が顧客の ID を航空会社に伝播する場合，航空会社のポリシーを取得し，航空会社が処理可能なトークンの種類やドメインを確認する．この内容と SIM の Mapping 要素に書かれているトークンやセキュリティドメインが同じであれば，航空会社に ID を転送することが可能であることが確認できる．

セキュリティポリシーを生成するには，アプリケーション開発者は提供された callerType の中から，必要な callerType を選択して注釈をつけるだけでよく，セキュリティポリシーは注釈からのモデル変換により得られる．注釈でアプリケーション依存の情報だけを指定し，そのほかに必要となるプラットフォームに関連した情報は SIM から取得することで，アプリケーション開発者はセキュリティ設定のためにプラットフォームの詳細を知る必要はない．このように，ID 伝播のような複雑な認証プラットフォームにおいても，SIM を適切に表現することで，アプリケーション開発者にセキュリティに関する詳細な知識や経験を強要せずに，プラットフォームに対応したセキュリティポリシー生成が可能になる．

4. 具体例

ここでは具体例として，旅行代理店の SIM やモデル変換を示す．

図 7 は旅行代理店の SIM を表している．図 5 のような注釈と図 7 の SIM から，旅行代理店が顧客から受け取るメッセージのセキュリティポリシーと，航空会社に送るメッセージのセ

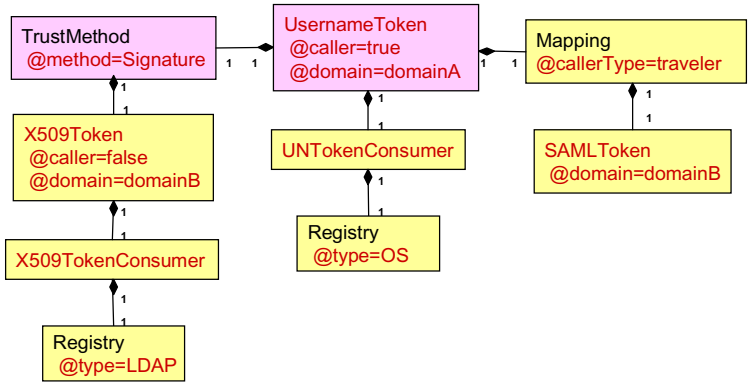

図 7. 旅行代理店のセキュリティ基盤モデル

キュリティポリシーを作成することを考える．

図 5 の注釈では callerType=traveler となっており，これに対応づけられているドメインは domainA であるとする．SIM には，caller 属性が true で，ドメインが domainA である UsernameToken 要素がある．したがって，これが旅行代理店の Caller すなわち顧客から受け取るトークンであることがわかる．この UsernameToken 要素には Mapping 要素と TrustMethod 要素が関連付けられている．Mapping 要素の callerType 属性は traveler であるので，traveler の ID は伝播する前に ID 変換を行う必要があることを示している．Mapping 要素の子要素は変換後のトークンの種類とドメインを示している．この例では，ドメインが domainA の UsernameToken をドメインが domainB の SAMLToken に変換されることを示している．

TrustMethod 要素は ID 伝播時に航空会社に対して自分の認証情報を送る必要があることを示す．この例では，TrustMethod 要素の method 属性が Signature であるので，旅行代理店の署名が必要であることを示している．

次に，セキュリティポリシーへのモデル変換について示す．セキュリティポリシーは，モデル上の注釈と SIM から 3.3 節のモデル変換によって生成される．顧客から受け取るメッセージに対するセキュリティポリシーは，SIM の UsernameToken 要素をセキュリティポリシーテンプレートに埋めることで生成できる（図 8）．航空会社へ ID 伝播するメッセージのセキュリティポリシーでも，Caller である UsernameToken 要素を参照する．ここで，UsernameToken 要素には Mapping 要素が関連付けられているため，顧客から受け取った domainA の UsernameToken をそのまま伝播するのではなく，domainB の SAMLToken に ID 変換して伝播する必要があることを示している．また，UsernameToken 要素には TrustMethod 要素も関連づいており，この method 属性は Signature であるので，航空会社が旅行代理店に対し署名を要求していることがわかる（図 9）．

```
<Authentication>
 <CallerToken>
  <SecurityDomain domainName="domainA"/>
  <UsernameToken>
   <WssUsernameToken10/>
  </UsernameToken>
 </CallerToken>
</Authentication>
```

図 8. 旅行代理店が受信するメッセージのセキュリティポリシー

```
<Authentication>
 <CallerToken>
  <SecurityDomain domainName="domainB"/>
  <SamlToken>
   <WssSamlV10Token10/>
  </SamlToken>
  <TrustMethod method="Signature"/>
 </CallerToken>
</Authentication>
```

図 9. 旅行代理店が送信するメッセージのセキュリティポリシー

旅行代理店が受信するメッセージのセキュリティポリシーと送信するメッセージのセキュリティポリシーは，それぞれ独立に生成されるが，セキュリティドメイン間の関連など認証プラットフォーム全体を SIM にモデル化したことで，互いに整合性のあるセキュリティポリシーが生成できる．認証プラットフォームが提供しているセキュリティ機構にかかわらず，アプリケーションレベルで注釈を指定するだけ

で各プラットフォームに適した正しいセキュリティポリシーが生成できることが大きな貢献といえる.

5. 関連研究

[4]は UML に付加したセキュリティアノテーションを使ったセキュリティ検証のフレームワークを提案している. セキュリティの設定に PIM を利用するという点では共通点があるが，[4]はセキュリティのプロトコルの検証を目的としている点で明確に異なる. SecureUML [5]は本研究と同様にモデル駆動型開発を用いたセキュリティ設定のための開発ツールを提供している．大きく違う点は，[5]は role-based access control (RBAC) の設定を目的としていることである.

我々の以前の研究 [6]では，MDS によるモデル変換対象は署名・暗号化用のセキュリティポリシーであった．また，注釈の代わりにセキュリティポリシーテンプレートを選択しなければならず, 認証プラットフォームをアプリケーション開発者が理解している必要があった. [7]ではさらにセキュリティ基盤モデルを署名・暗号化用に導入した．これらと比較して，本研究の新規性は，ID 伝播のような複雑な認証シナリオに対して SIM を定義し，注釈として抽象的なキーワードで認証を指定できるようにしたことである. その結果，アプリケーション開発者はプラットフォームを考慮することなく認証設定ができ, SIM を参照してセキュリティポリシーが生成できるモデル変換を提案できた.

6. まとめ

本研究では，MDS フレームワークによる認証設定の方法を提案した．これまでは，認証設定を行うには, 複雑な認証プラットフォーム全体をよく把握しておく必要があった. この問題を解決するために，本研究では，アプリケーションモデル上でセキュリティ設定を行うための注釈と, 認証プラットフォームをモデル化したセキュリティ基盤モデルを導入した. プラットフォームに依存しない抽象的な表現の注釈でセキュリティ設定を行い, 複雑な認証プラットフォーム全体の整合性を考慮して定義した SIM を参照することで, 認証プラットフォームの詳細を意識することなくセキュリティポリシー生成が可能になった.

今までのセキュリティ設定は認証プラットフォームの理解が必要だったことに比べて, 本研究での手法により, アプリケーション開発者でも簡単に認証設定が行えることは非常に大きな貢献である．また，アプリケーションを他のプラットフォームにデプロイする場合でも，そのプラットフォームの SIM が正しく設定されていれば, 注釈を変更することなくセキュリティポリシーを生成できる．これは，プラットフォームを限定しない SOA に非常に適した方法だといえる.

これまでの研究では，WS-Security に定義されている署名・暗号化・認証の 3 つをセキュリティ設定の対象としてきた．しかし，ほかにも考えなければならないセキュリティはいくつも存在する．たとえば認可，アクセス制御，プライバシーなどである. 今後はこれらもあわせて MDS フレームワークを使って設定できるようにしていきたい.

参考文献

[1] Web Services Security: SOAP Message Security 1.1, 1 February 2006. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf

[2] David S. F, *Model Driven Architecture: Applying MDA to Enterprise Computing*, John Wiley & Sons, 2003.

[3] Web Services Security Policy Language, ftp://www6.software.ibm.com/software/developer/library/ws-secpol.pdf

[4] Jan J¨urjens, “Sound Methods and Effective Tools for Model-based Security Engineering with UML”, *Proc. of the 27th Int.Conf. on Software Engineering*, St. Louis, 322, 2005.

[5] T. Lodderstedt, D. Basin, and J. Doser, “SecureUML: A UML-Based Modeling Language for Model-Driven Security”, *Proc. of the 5th Int. Conf. on The Unified Modeling Language (UML2002)*, Dresden, 2002. Lecture Notes in Computer Science vol. 2460, 426, 2002.

[6] M. Tatsubori, T. Imamura, Y. Nakamura, “Best-Practice Patterns and Tool Support for Configuring Secure Web Services Messaging”, *Pro. of Int. Conf. of Web Services 2004*, San Diego, 244, 2004.

[7] Y. Nakamura, M. Tatsubori, T. Imamura, and K. Ono, “Model-Driven Security Based on a Web Services Security Architecture”, *Int. Conf. on Service Computing 2005*, Orland, 2005.