Rational Software Architect
2004 年 6 月

サービス指向アーキテクチャーにおけるセキュリティー問題のモデリング

概念
多くの企業が Web サービスを使用するサービス指向アーキテクチャー (SOA) を導入し、モデル駆動型アーキテクチャー (MDA) の原則に従って、サービスを設計しています。MDA を表す際に用いられる UML に、ビジネス・プロセスで求められるセキュリティー・ニーズを示すモデル要素が欠けているため、システム・アーキテクトはモデルのセキュリティー問題を無視することを余儀なくされたり、あるいは、実装固有の方法で目的を示すことを余儀なくされています。本書では、ビジネス・ユーザーおよびソフトウェア・アーキテクトが UML 要素に適用することで、ビジネス利害関係者と協力してビジネス要件を把握できるステレオタイプとして、セキュリティーに関する目的要素を示す UML の有効なプロファイルを提案します。ここで提案する 1 つのプロファイルを使用することで、シンプルな振る舞いモデルの実装固有の詳細に対する MDA 制限に反することなく、設計段階でのセキュリティーのビジネス目的をアーキテクトが明確に示すことができます。

Simon Johnston
Architect
IBM Software Group

目次

概要

サービス指向アーキテクチャー (SOA) は、公開されている利用可能なインターフェースを通してアプリケーション、あるいはその他のサービスへのサービス提供を行うソフトウェアを設計する 1 つの方法です。各サービスでは、結合性の低い (多くの場合、非同期) メッセージ・ベースの通信モデルを通じて、多くの独立したビジネス機能を提供しています。SOA を利用するシステム・アーキテクトは、こうした 1 つ以上のサービスをコンポーネントとしてアプリケーションに組み込むことができます。

これまで、ソフトウェア業界の多くは、Web サービスとそれらの相互作用を実現するために基礎となる技術に注目していました。しかしながら、Web サービスを使用する企業規模のソフトウェア・ソリューションの構築に必要な技術やツールには、注意が十分払われていません。他の複雑な構造を持つ高品質のソフトウェア・ソリューションの設計には、設計、技術、構造パターン、スタイルを熟知したサポートによる早期段階での構造上の意思決定が必要になります。こうした方法を取ることで、拡張容易性、信頼性、およびセキュリティーなど、サービスに共通する問題に対処できます。[1]

ビジネス利害関係者は、ビジネス要件へのソリューション提供を IT 組織に依存しています。財政および市場重視の両方の理由から、利害関係者は IT ソリューションを提供するために必要な時間と資金を削減したいと考えています。

利害関係者はまた、各ソフトウェア・プロジェクトで提供される要件の範囲を最大にすることで、IT ソリューションから得られる価値を高めたいと考えています。こうした多くのプロジェクトには今日 Web サービスが必要であるため、SOA を使用して迅速で適切なビジネス要件を実現するためのより有効なツールと技術を備えることが非常に重要です。特に重要なのが、問題[2] を判別し、その問題に対する一貫した見識を示すモデリングです。複数の組織で運用するアプリケーションが多数あるため、サービス実装におけるセキュリティーは大きな問題です。本書の目的は、ビジネス利害関係者が要件のプロセス内でセキュリティーの目的を示すことができる一連の基本的なモデリング要素を提供することです。

アーキテクチャー・モデルと実装モデル

Web サービスを使用するアプリケーションを提供することが、IT の専門家に早急に求められています。多くの場合、アーキテクチャー・モデル (SOA) と実装モデル (Web サービス) を同時に提供する必要があります。あらゆる状況下で求められるため、モデルと実装の区別が付かなくなる場合もあります。本書では、モデル化された振る舞いを実装する際に用いられる技術とプラットフォームを、アプリケーションのアーキテクチャーや振る舞いのプラットフォームに依存しないモデルと決して混同しないように、モデル駆動型アーキテクチャー[3] ・アプローチを使用することを前提にしています。システム・アーキテクトは、ドメイン特化言語、または、サービス・ドメインの問題をモデル化する統一モデリング言語 (UML)[4] のプロファイルのいずれかを採用します。原則として、アーキテクトは、プラットフォームや言語の問題とモデルとを区別して考える必要があります。また、実装固有のセキュリティー問題についても同様です。例えば、サービスやメッセージの抽象的な概念を含むモデルには、サービス認証およびメッセージ署名を実装するための公開鍵暗号と証明書をメッセージがどのように使用できるのかの詳細は含まれません。これを含むと、特定の技術的な実装の詳細をプラットフォームに依存しないモデルに組み込むことになり、基本原則 (問題を区別する必要性) に反することになるからです。また、セキュリティーを軽視することはできません。セキュリティーの実装は複雑です。パフォーマンスに重大な影響を与え、サービスをサポートする IT インフラストラクチャーに対する要件を増やす場合があります。そのため、セキュリティー上の問題を他の問題と同じように慎重にモデル化することを誰もが最優先にするわけです。

IT 組織が問題を判別することで、ビジネス利害関係者は要件の範囲を最大にし、ビジネス・ニーズを理解して明確に示すことができます。ここでは、MDA の原則に従い、振る舞いの問題と実装やプラットフォーム固有の問題とを区別しつつ、シンプルなモデルでセキュリティーの目的を明確にする方法を説明します。

セキュリティー上の問題の認識

B2B 標準の RosettaNet[5] ファミリーを開発したチームで、ここで触れた内容に似た問題が生じ、その対応を始めました。ビジネス・アーキテクト、つまり、RosettaNet チームがデータや処理要件を収集した主要な利害関係者に対して、簡単にまとめた選択肢を示したわけです。ビジネス・アーキテクトはセキュリティー問題の技術的な詳細には詳しくありませんでしたが、セキュリティー対策を施さずに送信される可能性のあるデータと、安全な方法で送信する必要のあるデータを区別することはできました。

ただし、このアプローチには 1 つ問題がありました。それは、簡易化した用語が必要だったことです。用語が複雑であったり、あいまいな場合、ビジネス利害関係者は安全のため、使用可能なあらゆる種類のセキュリティーを要求します。次善の設計を求めてくる場合もあります。これは、ユーザーにとって理解しやすい説明と制約の仕様を簡単なガイドラインで示すようアーキテクチャー・グループに求める利害関係者がいることを意味しています。利害関係者は必要な見識を得ることで、コストや利益に関する十分な判断が可能になります。例えば、RosettaNet チームは事例を用いて、データの暗号化に要するコストが保護対象データの価値を大幅に上回る場合、ビジネス・ユーザーがそれを把握できるようにサポートを行いました。

セキュリティー問題の汎用化

一般的なソフトウェア・セキュリティーの問題に関するテキストは多数あります。特定のセキュリティーの実装や技術に関するものはさらに多数存在します。それでも、セキュリティーに関する技術的な実装のきっかけになる根本的な目的を追求したいと考えています。特に、わかりやすく、特定の技術的な実装の判断に使用できる基本的な目的について説明したいと考えています。

「セキュリティー」という言葉に含まれる根本的な問題とは何でしょう。一般的な例を用いて考えてみましょう。例えば、ATM 機から現金を引き出す場合、まず、ATM 機に向かうと、次の 2 つを要求されます。

自分のキャッシュ・カード (正式な ID となる) と暗証番号 (PIN) (自分や利用している銀行にとっては「共有の秘密」だが、他人には知られていない番号) です。

ここで ATM は入力された識別情報と暗証番号を銀行に照会し、ATM の前に立っている人間が、口座名義人本人かどうかを信頼できるレベルで確認します。カード発行元の銀行が入力された詳細内容を承認した場合、口座名義人の追加情報を提供するセキュリティー証明書の一式が ATM に送信されます。この口座固有の情報に基づいて、ATM は承認された口座名義人が実行可能な操作の一覧を画面に表示します。この操作には、「引き出し」や「預け入れ」などが含まれます。実際に選択できる操作内容は、以下の 2 つの項目に共通した部分です。

- この ATM が実行可能な一連のすべての操作
- カード発行元の銀行が、口座名義人が実行可能と認定している一連のすべての操作

銀行が送信する証明書には通常、一度に引き出せる金額の上限 (ATM 側で指定可) が含まれます。ATM のシステム・アーキテクトは、ATM によって入出力されるすべての情報を記録することで監査証跡を提供することも必要です。

銀行と ATM 間のこうした通信はどのように行われているのでしょうか。銀行は ATM から得られる情報を、また ATM は銀行から得られる情報をどうすれば信用できるのでしょう。セキュリティー・プロトコル、データ暗号化などに関するこうした技術的な詳細は、重要であり、興味深いものでもあります。ただし、その内容は明らかにシンプルな振る舞いモデルの適用範囲外です。

ATM の例を次の 3 つのセキュリティー上の問題または領域に分けて説明します。

- 識別(識別、認証)
- 操作(承認)
- 非公開(プライバシーの保護)

4 つ目の領域はあまり認識されていませんが、他の 3 つの領域に関係しています。

- 結果(監査)

この監査領域は、多くの場合、多数の IT アプリケーション開発分野で軽視されがちです。また、コア・セキュリティーなどの一部のビジネス分野や、EDI (電子データ交換) などのアプリケーションでは、規制上の問題に対してかなりの監査能力を要するため、監査機能は明らかに重要なセキュリティー問題と言えます。監査を暗黙的な目的とみなすことで、結果として、提供する基本要素がすべての詳細な振る舞いの監査証跡となります。例えば、当事者 A が当事者 B と提携する前に、当事者 B を認証する必要があるとします。これは、他の実装の詳細と同様、日時を含む認証要求や対応内容をすべて監査する必要があることを意味しています。

図 1 は、これらの領域での依存関係を示しています。例えば、認証を行わないと承認はできません。また、承認と認証には共に監査が不可欠であり、実装の場合を除き、分析と否認防止の例外を把握することが必要です。

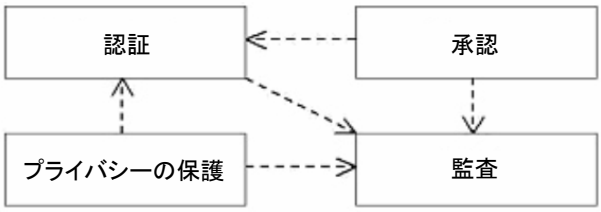

図 1: セキュリティー領域間の依存関係

本書では、これらの領域で共通の基本目的に対応する方法をさらに提供します。また、モデルに組み込んだ基本要素がどのように特定の技術における実装のきっかけになるのかを説明します。

識別

この領域は主に、交流のある、または協力関係にある 1 人以上の当事者の識別に関するものです。
実際には、この領域は 2 つの問題に分けられます。識別という静的概念と認証という動的概念です。ATM の例では、ATM カードは静的な銀行発行の ID ですが、カードと PIN によって、ATM は口座名義人を動的に認証することができます。識別よりも認証の概念をモデル化することがはるかに重要です。一般的に、識別は認証よりもはるかに実装の技術的な詳細に関係しています。このため、例えば、当事者間の特定の協力関係において、当事者間の認証が必要だと非常に簡潔にいうことができます。これは明示的にも暗黙的にも表現できます。例えば、信頼という別の概念は、信頼領域を説明する場合に使用できます。同じ信頼領域内に存在する関係者は、互いを認証する必要はありませんが、信頼領域の境界線を越える機密情報を扱うような場合は、認証が必要になります。

信頼領域の概念は実用的な例ですが、すべてのニーズに当てはまるわけではありません。多くの場合、信頼領域には階層があり、外部の関係者との通信には、認証が不要な場合もあります。

ファイアウォールの内側であれば、社員以外の誰もネットワーク・リソースにアクセスできないという点では、ビジネス自体を 1 つの信頼領域と考えることができます。しかし、ERP (Enterprise Resource Planning: エンタープライズ・リソース・プランニング) のアプリケーションと CRM (Customer Relationship Management: カスタマー・リレーションシップ・マネージメント) のアプリケーションを区別した信頼領域が存在することがよくあり、この場合、このように単一領域として考えることはできません。それは、この 2 つのアプリケーションがそれぞれ独自のアプリケーション・レベルでセキュリティーを実装し、維持しているためです。また、ビジネス自体を、トレーディング・エクストラネットのメンバーシップにより、外部の信頼領域のメンバーと考えることもできます。この点については、信頼と認証にはそれぞれ、重複する部分もありますが、コラボレーション・モデルに適用できる明確な目的があると言えます。

操作

この領域は、主に、本人確認後、実行を承認された操作内容のみにユーザーのオプションを制限できるようにします。この領域では、認証機能が不可欠です。ユーザーに対して操作を許可する前に本人確認が必要だからです。ビジネス利害関係者はこの機能が安全であるという確証を求めます。つまり、ユーザーを識別する機能や (監査を通じて) 操作日時を把握する能力が求められているわけです。もちろん、中には承認を必要としない機能もありますが、それは誰にでも利用できる内容である場合か、パフォーマンス上の理由から、信頼領域にあるサービスで、要求者のアクセス権が確実に想定されていると認識している場合です。

この信頼領域の概念は、パフォーマンスを別の問題、例えば、競合の問題として見た場合に重要になります。セキュリティーの実装に関するコストが生じ、場合によってはかなり高いコストとなるためです。お客様の未処理注文の数を返す機能 (データベースからの非常にシンプルな照会) について考えてみましょう。認証、承認、プライバシーの保護 (以下を参照) を求めれば、次のような場合に莫大な費用がかかることになります。

- 要求者に対する証明書の提出依頼
- 通常、リモート・サービスによる、提出された証明書の確認
- 返された情報の暗号化

要求者や提供者のアプリケーション・サービスが同じである場合は、1 つの信頼領域として識別することができます。そうすることで、費用を抑えつつ、パフォーマンスを強化して利益を得ることができます。アーキテクチャーからの視点では、信頼領域間で行われるすべての通信を理解することも重要です。この通信は、可能な限り最小化し、制御する必要があります。セキュリティーの実装全体において障害 (あるいは外部からの攻撃) が発生する可能性のもっとも高い領域だからです。

実装について、次の主な 2 つの興味深い承認方法があります。

- **関係者の個別承認**。機能への一連のアクセス権を各関係者に明示的に割り当てることができます (ただし、プロセスを最適化するために、明示的なアクセス権がない場合、アクセス権が暗黙的に許可されているか、あるいは許可されていないものとして対応することがあります)。
- **役割による承認**。さまざまな役割をアプリケーションごとに作成し、アクセス権を上記のように、個別の関係者にではなく、役割に対して割り当てることができます。各関係者が認証されると、関係者に与えられた証明書に関係者の役割が記載されます。その役割により、特定機能へのアクセスを承認された関係者であるかどうかが判別されます。

ここで大事なことは、シンプルな振る舞いモデルを考える際に、常に上記のような詳細を含まないことです。シンプルなモデリング段階での目的は、単に注意を促すことで、例えば、特定の機能を使用するには実行前に承認が必要であると説明しているだけで、承認を得る詳しい方法については触れていません。

非公開

プライバシー保護領域の問題とは、承認された本人のみに限定された内容の情報を、承認されていない他人に提供しないことを保証することです。ビジネスの世界では、莫大な量のデータが消費され、生成されています。そのデータは保管され、処理され、手渡されて、ビジネスの運営をサポートします。こうした機密情報を保護し、情報やサービスの要求者または提供者の識別方法を提供する必要があります。つまり、要求されている、あるいは、提供されているサービスを単に把握するだけでなく、どの認証エンド・ユーザーがそのサービスを介する取引相手なのかを把握している必要があるのです。

プライバシーの領域に関連した、次のような明らかに異なる 2 つの目的があります。

- メッセージまたは文書への署名 (複数の署名を使用する場合があります)。これには、メッセージまたは文書を作成したエンド・ユーザーまたはサービス (共に複数可) を識別する目的があります。この目的は、企業間の商取引、政府機関との商取引、電子メールでの企業通信にも幅広く使用され、電子署名の多数の共通規格に依存しています。文書には、複数の署名が用いられている場合があります。例えば、注文が行われると供給依頼書には、発行者、管理者 (承認者)、最終的には購買部門が署名し、すべての署名は文書に明記されます。
- メッセージの確実なプライバシー保護という目的。これには、信頼できる媒体を介した送信、あるいは、メッセージ内容の暗号化またはセキュリティー保護が含まれます。この目的は、実装の選択がもっとも多い領域と言えるでしょう。例えば、電子メッセージがサービス間で転送されたとします。メッセージ自体は送信者によって暗号化されていても、安全ではないチャネルを介して送信されたかもしれません。また、メッセージは、HTTPS や TLS (どちらもサービスの一環として暗号化が行われるが、送信者による制御は不可) などの安全な方法で送信されているが、非暗号化テキストとして送信されたかもしれません。あるいは、メッセージに、その他の安全な方法 (コピー防止用紙で印刷され、クーリエ便で送付) により送られた文書の ID が含まれているかもしれません。

別の問題がデータ保存性の領域でよく見られます。データ保全とは、メッセージまたは文書の内容が転送中にさまざまな取引相手によって改ざんされるのを防止することです。データ保全性を備えたメッセージまたは文書とは、改ざん防止機能が付いたものを指します。メッセージがプライベートな内容である場合、この改ざん防止機能を用いることが望まれます。ただし、改ざん防止機能を用いたメッセージでもプライバシー・ソリューションを必要としないものもあります。

実装者の役割は、最終的にデータの機密性に関する要件を満たすソリューションをビジネス分野で定められたガイドラインと共に利用できるようにすることです。

例えば、お客様のクレジット・カード番号を銀行に転送して確認する際、クーリエを使うことは e-commerce の Web サイトでは許可されません。クーリエは政府の極秘文書を配信する場合に最適な方法です。

否認

この領域は、発信元や内容の否認防止の概念に関するものです。この用語は多くの EDI 文書で見られます。否認防止は、将来、取引相手の 1 人が取引の完了を拒否した場合に、重要な意味を持ちます。あるいは、取引相手の 1 人が、取引が行われたことは認めるが、その具体的な詳細について異論を唱える場合もあります。購入した 100,000 株の価値が下がり、当事者 A が取引では 100 株だけだったと主張する場合もあります。多くの業界や地域では、このような状況に対処するため、取引記録を長期間保持するように法的に規定しています。

この件については、前述の監査の問題が、必要なメッセージ保管の機能を提供します。監査だけでは、否認防止の機能としては十分とは言えません。認証 (発行元証明) を含む、交換されるメッセージの監査 (内容証明) は通常、証明に必要なだけのレベルで行われるからです。

モデルへの基本要素の適用

以下に、推奨するプロファイル (詳細は、本書の「プロファイルの詳細」セクションを参照) を使用して、2 人の当事者間で交換される文書のシンプルなモデルでセキュリティーの目的を説明する方法の例を示します。以下の表は、例で使用する目的要素の要約です。

目的	*注釈*
監査	特定の通信に監査が必要である場合に使用します。監査にはすべての当事者の認証された身元証明と当事者間で通信される任意のデータの両方を含むと想定されます。
認証	特定の協力関係の適用範囲において、当事者に認証が必要な場合に使用します。
承認	2 人の当事者間の通信において、要求者に要求を行う許可が必要な場合に使用します。
非公開	該当する情報がプライベートな内容である場合や、データがプライベートな内容 (不正アクセスからの保護が必要なもの) であり、改ざん防止機能付き (改ざんされずに宛先に届くことが保証されているもの) であることを保証するために妥当な行動 (技術的な意味を含む) が必要な場合に使用します。
署名	文書に関係する当事者の電子署名が、該当する情報に含まれる場合に使用します。
改ざん防止	当事者間で転送されるデータが、送信者が発信したときと同じ形式で、同じ内容と意味を持つデータが受信者の元に届くことを保証する必要がある場合に使用します。

目的	*注釈*
信頼	協力関係にある当事者が明らかに信頼領域に存在する場合に使用します。

以下は、UML のアクティビティー・モデル例を使用して、買い手と売り手との間で交わされる発注書について説明したものです。この例では、目的要素を一般的なモデル要素に適用する UML のステレオタイプとして認識しています。

図 2 はこの協力関係の 3 つのセキュリティー目的を示しています。

- 買い手は売り手に認証される必要があります。
- 「受注」アクションには承認が必要です。
- 発注書のオブジェクト・フローには署名が必要です。改ざん防止が必要な場合もあります。

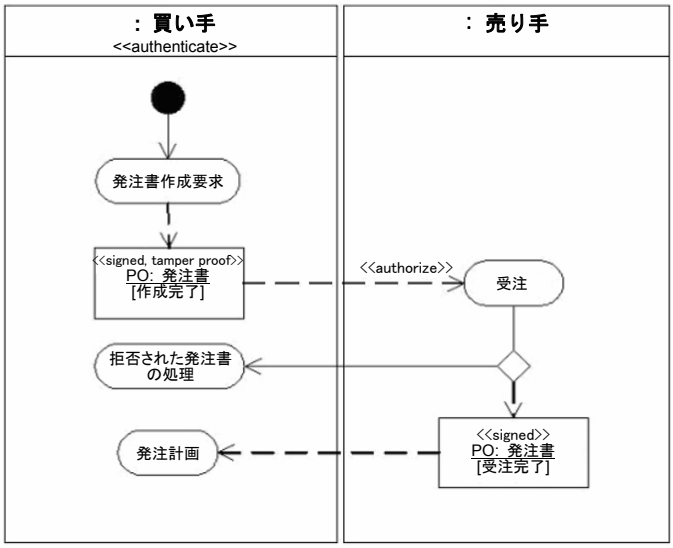

図 2: 購買取引におけるセキュリティーの目的

モデルには、サービス、エンドポイント、インターフェース、スキーマ、XML などの技術的な項目は含まれていません。これは、ビジネス利害関係者からセキュリティーに関する要件を引き出すための手段として推奨するシンプルな独自のプラットフォームです。

本書の「プロファイルの詳細」セクションで説明するプロファイルでは、ユーザーがプラットフォームに依存しないモデルを開発する際に使用したいと考えるモデリング・ツールまたは方法についてはほとんど触れていません。例えば、図 2 にはビジネスの振る舞いをモデル化するためのアクティビティー図を使用し、図 4 ではシーケンス図を使用しています。また、コラボレーション図や状態マシンを使用して、ビジネス・アプリケーションの振る舞いをモデル化することもできます。

基本要素から実装へのマッピング例

次に、上記で説明した一部の基本要素から特定の技術設計へのマッピングで考えられる例について説明します。実際に選択可能な実装については、本書の後半で説明します。

前述のとおり、MDA はシンプルなモデルをモデル間の変換により実装固有モデルに変換する方法です。以下に示すパターンに組み込まれています。

プロトコルとパターン

まず、技術的な実装を表現する方法が必要です。UML では、コラボレーションのテンプレートを使用して、パターンを表します。このパターンは、特定のモデル要素に詳細を加えて拡張するためのものです。ここでは、モデルのそれぞれの目的に応じて技術固有の実装を表すパターンを作成する必要があります。

図 3 は、信頼できる検証サービスを用いる承認方法のパターンを示しています。IT グループには、異なる実装、または場合によってはパフォーマンスなどの別の特性に使用する複数の選択可能なパターンが存在する可能性があることに注意してください。ここでは、要求者のオブジェクト、承認対象の方法、使用する検証サービスなどの 3 つのパラメーターを使ってパターンを説明します。図 3 では、信頼領域 (提供者および検証者のオブジェクトを囲む枠内) も示しています。検証サービスが承認された方法の提供者に信頼される必要があることを、図を使って視覚的に注意を促すものです。これは大事なことです。信頼領域内では、パフォーマンスを最適に活用するため、頻繁に対話が行われる可能性のある 2 つのサービス間におけるセキュリティーの実装を省いてしまうからです。

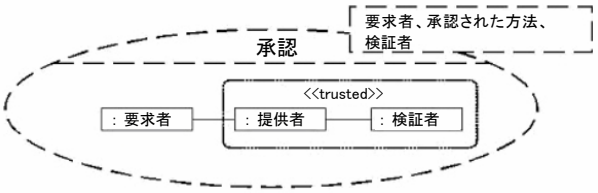

図 3: 承認パターン

このパターンでは、実装の振る舞いについて説明しています。図 4 では、提供者が認証者を呼び出して要求者の証明書を検証する方法や、認証者から返される内容に応じて、その方法を実行するか、承認の例外を示す方法を表しています。

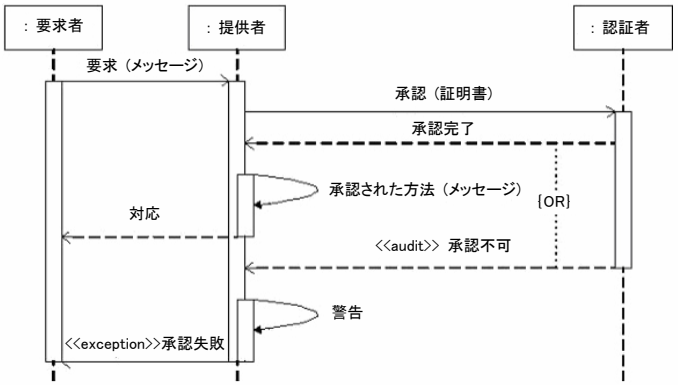

図 4: 実装の振る舞い

この例では、図 3 で定義した 3 人の当事者間で発生した一連のイベントを示す UML メッセージのシーケンス図を使用しています。

図 4 にはステレオタイプが追加されています。特に、承認に失敗したことを意味する検証サービスから返されたメッセージが監査対象になっていることに注目してください。

繰り返しますが、ほとんどの明示的な目的には監査が必要です。図 4 を見ると、認証者が「NotAuthorized (承認不可)」を返した場合、ステレオタイプの監査の対象になることがはっきりとわかります。理由は、このイベントを監査することはビジネスにおいて必要なこと(任意の黙示的なセキュリティー上の要件を含む) だからです。
ステレオタイプの例外は、主要な UML 仕様の一部であり、推奨するプロファイルの一部ではありません。また、制約 {OR} を使用して、考えられる 2 つの承認結果を 1 つの図にモデル化していることに注意してください。

パターンとアクティビティー・モデル例を結び付けるには、図 5 のように、パターンのパラメーターをモデルの要素に置き換える必要があります。売り手オブジェクトは要求者に、受注処理アクションは保護された方法、XWSKeySvr と呼ばれるサービスは検証者に置き換えます。モデルにパターン事例を作成するためです。こうすることで、プラットフォームに依存しないモデル内の「受注」などの多くの異なるメッセージ事例に同一のパターンを結び付けて考えることができます。

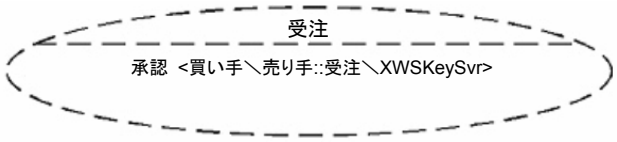

図 5: パターンのモデルへの結合

この結合はモデル内で継続して実行することができます。したがって、後日 (例えば、サービス間の検証者を変更する場合など)、結合に手を加えて別の実装を作成することができます。

実装の選択

承認あるいはプライバシーの保護などの目的で実装を行う際に使用する設計パターンの作成には、多数の設計方法やプラットフォーム固有の技術的なソリューションがあります。例えば、上記でも説明しましたが、承認を実装する場合の当事者と役割を区別する考え方があります。実際に、これはとても一般的な方法であり、J2EE や Microsoft .NETなどのミドルウェアでも使用されています。本書では、こうした設計上の決定やパターンの詳細に重点を置いていませんが、特定の SOA パターンを、Gangof-Four[6] などのさまざまな共通パターンや、サービス指向アプリケーション・インフラストラクチャーの技術制約と実態に固有のまったく新しいパターンとして認識しています。

標準は、SOA の採用や実際の形状に大きく影響します。World Wide Web Consortium (W3C) や Organization for the Advancement of Structured Information Standards (OASIS) などの基本標準が基礎となります。RosettaNet に基づく業界標準は、内部的にも外部の関係者と連携して行うビジネスにも深く関係するコンテンツやプロセスを提供します。ただし、楽観的に考えすぎないよう注意が必要です。こうした標準の多くは比較的新しく、その開発には多数の商業的な利益や学究的な関心があり、標準を構成するための標準 (WS-1) を管理するために設立された組織さえ存在します。

以下の表は、Web サービス領域のセキュリティーに関連する現行仕様のスナップショットです。

これらの仕様には、それぞれの仕様間の深いつながり、および、これらの仕様と基準となる W3C や OASIS XML の仕様との関係は別として、複雑で、当然、変更される可能性があるという問題が存在します。

XML	*メッセージング*	*セキュリティー*	*関連*
XML	SOAP	XML Encryption	WS-Policy
XML Namespaces MTOP	WS-Security	WS-PolicyAssertions	
XML InfoSet	WS-Addressing	WS-SecureConversation	WS-PolicyAttachment
XInclude	WS-Routing	WS-Trust	WS-SecurityPolicy
XPath		WS-Federation	XML Query
		Active Requestor Profile	
		Passive Requestor Profile	
		Web Services Security Kerberos Binding	
		Web Services Security Kerberos Binding	

サービス指向アーキテクチャーと Web サービスの実装は、多数の IT 組織にとって、統合のチャンスを広げる大きな利益になると考えられます。Web サービスを使用した既存のアプリケーションの再構築を試みる際は、慎重にモデル化し、その内容を完全に理解することが必要です。また、すべての主要な問題をビジネス利害関係者との協力関係と区別して処理する必要もあります。

プロファイルの詳細

このセクションでは、ビジネス利害関係者の要件を把握する際の UML 要素に適用可能なステレオタイプの目的要素を表す UML の推奨プロファイルについて説明します。

ステレオタイプの監査

メタクラス

ActivityNode、Message

説明

特定の通信に監査が必要である場合に使用します。監査にはすべての当事者の認証された身元証明と当事者間で通信される任意のデータの両方を含みます。

監査は、任意の通信のステレオタイプの「承認」においては暗黙的に行われますが、認証の実装や署名付きのプライベートなデータを例外処理では明示的に行われます。

ActivityNode に適用する場合、アクティビティー図内の操作、構造化アクティビティー、および制御ノード (決定など) に注釈を付けることができます。相互作用の Message に適用する場合は、表示されているモデル要素間で送信するメッセージに注釈を付けることが可能です。

プロパティー

なし。

注釈

必要な注釈はありません。

ステレオタイプの認証

メタクラス

ActivityPartition、Lifeline

説明

特定の協力関係の適用範囲において、当事者に認証が必要な場合に使用します。このステレオタイプは、協力関係にある当事者の事例を表す振る舞いモデルの要素に適用します。

ActivityPartition (アクティビティー図の場合) または Lifeline (相互作用図の場合) のいずれかに適用する場合は、表示されているモデル要素に注釈を付けることができます。

プロパティー

なし。

注釈

必要な注釈はありません。

ステレオタイプの承認

メタクラス

ActivityNode、Message

説明

2 人の当事者間の通信において、要求者に要求を行う許可が必要な場合に使用します。このステレオタイプは、振る舞いモデルのメッセージやフローに適用します。このとき、呼び出される振る舞いは認証確認によって保護されます。

プロパティー

なし。

注釈

必要な注釈はありません。

ステレオタイプの非公開

メタクラス

ObjectNode、Class

説明

通信で転送されるデータがプライベートな内容である場合や、データの改ざん防止と保護を保証するために妥当な行動 (技術的な意味を含む) が必要な場合に使用します。

同じ要素に対してこのステレオタイプの改ざん防止と非公開を適用しないでください。非公開には改ざん防止の意味が含まれています。つまり、ステレオタイプの非公開を適用することは、データが非公開 (不正アクセスから保護されている) であり、改ざん防止機能付き (改ざんされずに宛先に届くことが保証されているもの) であることを意味します。一方、ステレオタイプの改ざん防止を適用することは、不正アクセスからの保護を行わなくても改ざんされることなく宛先にデータを届けると保証することを意味します。

Class (または UML の派生要素) に適用した場合、この要素が振る舞いモデルに存在するときは常に署名付きであることを意味します。要素が特別に処理される場合の振る舞いモデルの事例にステレオタイプを適用することもできます。

プロパティー

なし。

注釈

必要な注釈はありません。

ステレオタイプの署名

メタクラス

ObjectNode、Class

説明

通信で転送されるデータに当事者を証明する署名の何らかの概念が含まれている場合に使用します。ここで重要なのは、当事者が通信の際に要素に署名する必要はないという点と、複数の署名を付けることが可能だという点です。

プロファイルは誰の署名が必要なのかを指定するものでも、必要な署名の数を指定するものでもありません。

Class (または UML の派生要素) に適用した場合、この要素が振る舞いモデルに存在するときは常に署名付きであることを意味します。要素が特別に処理される場合の振る舞いモデルの事例にステレオタイプを適用することもできます。

プロパティー

なし。

注釈

必要な注釈はありません。

ステレオタイプの改ざん防止

メタクラス

ObjectNode、Class

説明

当事者間で転送されるデータが、送信者が発信したときと同じ形式で、同じ内容と意味を持つデータが受信者の元に届くことを保証する必要がある場合に使用します。

同じ要素に対してこのステレオタイプの改ざん防止と非公開を適用しないでください。非公開には改ざん防止の意味が含まれています。つまり、ステレオタイプの改ざん防止を適用することは、不正アクセスからの保護を行わなくても改ざんされることなく宛先にデータを届けると保証することを意味します。一方、ステレオタイプの非公開を適用することは、データが非公開 (不正アクセスから保護されている) であり、改ざん防止機能付き (改ざんされずに宛先に届くことが保証されているもの) であることを意味します。

プロパティー

なし。

注釈

必要な注釈はありません。

ステレオタイプの信頼

メタクラス

ConnectableElement

説明

協力関係にある当事者が明らかに信頼領域に存在する場合に使用します。

ConnectableElement はこの場合、協力関係での役割を表す一連の要素を意味しています (図 6 参照)。

プロパティー

名前	*種類*	*注釈*
領域	文字列	領域名と領域内の関係者は、同じ領域名を持つ特定のモデルの一連の要素になります。

注釈

コラボレーション図で領域の境界を明確にわかりやすく示す場合に便利です。この例は、承認パターンの詳細で示されています。図 6 のように、関係者の信頼領域の境界は点線で描かれています。

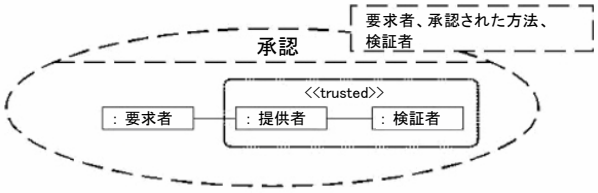

図 6: 信頼領域の境界

参考資料

[1] Brown, A.、Johnston, S.、および Kelly, K.、「*Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications*」、Rational Software White Paper

[2] Lopes, C.V. および Hursch, W.L.、「*Separation of Concerns*, Tech Report of College of Computer Science」、Northeaster University、Boston、MA (1995 年 2 月 24 日発行)

[3] OMG, MDA, *An Introduction*, OMG

[4] OMG, UML 2.0 *Superstructure Specification*, OMG

[5] RosettaNet Consortium [www.rosettanet.org]

[6] Gamma, E.、Helm, R.、Johnson, R.、および Vlissides, J.、「*Design Patterns, Elements of Reusable Object-Oriented Software*」、Addison Wesley

本書は、IBM developerWorks の Web サイトで公開された原著を転載したものです。

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America

March 2005.