Rational Software Architect

Junho de 2004

Preocupações com Segurança em Modelagem na Arquitetura Orientada a Serviços

Resumo

Muitas empresas estão implementando a SOA (arquitetura orientada a serviços), utilizando serviços da Web, e estão projetando esses serviços de acordo com os princípios de MDA (Model Driven Architecture). Como a UML utilizada para expressar elementos do modelo de MDA de deficiências para indicar as necessidades de segurança dos processos de negócios, os arquitetos de sistema são forçados a ignorar as preocupações com segurança em seus modelos ou indicar suas intenções de maneira específica à implementação. Esse documento propõe um perfil de candidato para UML que apresenta os elementos de intenção relacionados à segurança como estereótipos que os usuários de negócios e arquitetos de software possam aplicar aos elementos UML quando trabalharem com investidores cooperativos para capturar os requisitos de negócios. Utilizando um perfil como o proposto aqui, os arquitetos podem especificar a intenção do negócio de segurança em seus designs sem violar a proibição MDA contra detalhes específicos à implementação em modelos comportamentais de alto nível.

Simon Johnston

Arquiteto

IBM Software Group

Conteúdo

Introdução

Uma SOA é uma maneira de projetar software para fornecer serviços para aplicativos, ou outros serviços, por meio de interfaces publicadas e descobríveis. Cada serviço fornece uma parte discreta de funcionalidade comercial por meio de um modelo de comunicação desconexo (geralmente assíncrono) baseado em mensagem. Arquitetos de sistema que utilizam a SOA podem incorporar um ou mais serviços em seus aplicativos como componentes.

Até o momento, grande parte do foco do segmento de mercado de software tem sido a tecnologia subjacente para implementar serviços da Web e suas nterações. Atenção insuficiente foi dedicada às técnicas e ferramentas necessárias para arquitetar soluções de software em escala empresarial utilizando serviços da Web. O design de uma solução de software de alta qualidade, como qualquer estrutura complexa, requer decisões arquiteturais antecipadas suportadas por técnicas de design, padrões estruturais e estilos bem entendidos. Esses padrões tratam de problemas de serviços comuns, como escalabilidade, confiabilidade e segurança.[1]

Os acionistas comerciais dependem da organização de Tecnologia de Informação para fornecer soluções para os requisitos comerciais. Por motivos financeiros e de mercado, os acionistas desejam diminuir os investimentos em tempo e dinheiro gastos para fornecer soluções de Tecnologia de Informação. Também desejam aumentar a importância que derivam das soluções de Tecnologia de Informação, maximizando a cobertura de requisitos que cada projeto de software fornece. Como muitos desses projetos atualmente envolvem serviços da Web, é muito importante que tenhamos ferramentas e técnicas para a implementação rápida e bem-sucedida dos requisitos comerciais que utilizam a SOA. Consideramos a modelagem especialmente importante, devido à sua capacidade de separar questões[2] e apresentar uma visão unificada dessas questões. A segurança em implementações de segurança é a principal questão, porque muitos aplicativos operam entre limites organizacionais. O objetivo deste documento é fornecer um conjunto de elementos de modelagem primitivos que permitem que os acionistas comerciais especifiquem a intenção de segurança no processo de requisitos.

Arquitetura versus Modelos de Implementação

À medida que os profissionais de Tecnologia de Informação se apressam para fornecer aplicativos que utilizam serviços da Web, sempre se encontram na posição de acelerarem um modelo de arquitetura (SOA) e um modelo de implementação (serviços da Web) simultaneamente. Como é de se esperar sob essas circunstâncias, as distinções entre o modelo e a implementação algumas vezes são perdidas. Este documento considera o uso de uma abordagem de Model Driven Architecture[3], que evita cautelosamente misturar o modelo independente de plataforma da arquitetura de um aplicativo e o comportamento com as tecnologias e plataformas utilizadas para implementar esse comportamento modelado. Os arquitetos de sistema empregam linguagens ou perfis específicos ao domínio para a UML (Linguagem de Modelagem Unificada)[4] a fim de modelar as questões do domínio de serviço. Os princípios que requerem que os arquitetos separem as questões de plataforma e idioma desse modelo também requerem que separem questões de segurança específicas à implementação. Por exemplo, um modelo que inclua noções abstratas de serviços e mensagens também não deve incluir detalhes de como as mensagens podem utilizar criptografia e certificados de chave pública para implementar autenticação de serviço e assinaturas de mensagem, porque isso viola um princípio muito fundamental (a necessidade de separar questões), introduzindo os detalhes de uma implementação técnica específica no modelo independente de plataforma. Por outro lado, não é possível tratar a segurança como uma questão secundária. As implementações de segurança são complexas, e podem ter sério impacto sobre o desempenho, além de estabelecer requisitos adicionais para a infra-estrutura de Tecnologia de Informação que suporta os serviços. Portanto, é do maior interesse de todos modelar as questões de segurança tão cuidadosamente quanto qualquer outra questão.

Separando as questões, a organização de TI pode engajar com sucesso os acionistas comerciais na compreensão e descrição da necessidade da empresa de maximizar a cobertura dos requisitos. Pretendemos mostrar como especificar planos de segurança nesses modelos de alto nível, isolando as questões de comportamento da implementação e as de plataforma específica, consistente com os princípios do MDA.

Revisitando Questões de Segurança

A equipe que desenvolveu a família RosettaNet[5] de padrões B2B levantou questões semelhantes às que acabamos de discutir e começaram a tratá-las. A idéia era apresentar um conjunto simplificado de opções aos arquitetos de negócios — os acionistas principais de quem a equipe do RosettaNet reuniu os dados e os requisitos de processamento. Esses arquitetos de negócios não eram experientes nos detalhes técnicos das questões de segurança, mas estavam aptos a distinguir os dados que precisavam ser transmitidos de maneira segura daqueles que podiam ser enviados sem medidas de segurança.

Entretanto, um problema com essa abordagem era que uma terminologia simplificada era vital. Assim que os termos se tornassem complexos ou opacos, o acionista comercial solicitaria cada tipo de segurança disponível, apenas para estar seguro, e isso resultava em designs parcialmente ideais. Era esse o comportamento na parte dos acionistas que lideravam o grupo de arquitetura para estabelecer diretrizes simples sobre a especificação de tais restrições, bem como descrições que o usuário pudesse entender. Isso dava aos acionistas a percepção que eles precisavam para tomar decisões informadas do custo/benefício dos negócios. Por exemplo, a equipe RosettaNet utilizava exemplos para ajudar o usuário corporativo a entender quando o custo de criptografia de dados era muito maior que o valor dos dados sendo protegidos.

Generalização de Problemas de Segurança

Há muitos textos sobre problemas de segurança geral de software, e muito mais sobre determinadas implementações e tecnologias de segurança. Entretanto, queremos estar aptos a tratar o plano básico que conduz as implementações técnicas relacionadas à segurança. Especificamente, gostaríamos de poder especificar um conjunto de planos iniciais descritivos, fáceis de entender, que pudessem ser utilizados para identificar determinadas implementações técnicas.

Quais problemas e preocupações básicas o tempo "segurança" inclui? Deixe-nos dar um exemplo comum: sacar dinheiro de uma máquina ATM. Em primeiro lugar, quando eu me dirijo a uma máquina ATM, sou solicitado a fornecer duas coisas: meu cartão ATM (que funciona como identificação formal) e um PIN (número de identificação pessoal), que é um "segredo compartilhado" que apenas eu e meu banco, e ninguém mais, conhece. A ATM agora pode tomar as informações de identificação e o PIN e perguntar ao meu banco se a pessoa que está de frente para ela pode ser assumida, com um nível aceitável de confiança, como portadora da conta. Se o banco emissor aprovar os detalhes fornecidos, ele enviará em retorno à ATM um conjunto de credenciais de segurança que fornecerá informações adicionais sobre o portador da conta. Com base nessas informações específicas da conta, a ATM então exibe a lista de ações que eu estou autorizado, como portador da conta, a executar — ações que poderão incluir "saque", "depósito" etc. Observe que esse conjunto de opções representa na verdade a interseção de dois conjuntos:

- O conjunto de todas as operações que esta ATM específica é capaz de executar
- O conjunto de todas as operações que o banco emissor certifica que eu estou qualificado a executar

As credenciais enviadas pelo banco normalmente incluem um limite sobre o valor que eu posso sacar em cada transação individual - um limite que a própria ATM pode impor. A arquitetura do sistema ATM também deve fornecer uma trilha de auditoria, registrando todas as informações que circulam pela ATM.

Como é feita essa comunicação entre o banco e a ATM? Como o banco pode confiar nas informações que ele obtém da ATM e vice-versa? Detalhes técnicos como esses, relacionados a protocolos de segurança, criptografia de dados etc., são importantes e interessantes — mas esses são precisamente o tipo de detalhes que ficam de fora de qualquer modelo comportamental de alto nível.

O exemplo da ATM ilustra três categorias de domínios ou questões de segurança:

- Quem é você? (Identificação, Autenticação)
- O que você pode fazer? (Autorização)
- O que você, e outros, podem ver? (Privacidade)

Um quarto domínio é menos óbvio, mas está ligado aos outros três domínios:

 O que aconteceu? (Auditoria)

O domínio de auditoria muitas vezes tende a ser uma reflexão tardia no desenvolvimento de muitos aplicativos de TI. Por outro lado, em algumas áreas de negócios, como segurança de núcleo, e em aplicativos como EDI (Electronic Data Interchange), nas quais as questões normativas exigem recursos significativos de auditoria, a função de auditoria é uma questão de segurança explícita e importante. Nossa abordagem trata a auditoria como um plano implícito; conseqüentemente, todas as questões iniciais que apresentamos implicam em uma trilha de auditoria do comportamento detalhado. Assim, por exemplo, quando marcamos que a parte A exige que antes de poder colaborar com a parte B, ela deve autenticar a parte B, isso implica em que esse pedido e resposta de autenticação com a data/hora, bem como os demais detalhes de implementação, deverão todos ser auditados. A figura 1 demonstra as dependências entre esses domínios. Por exemplo, não é possível implementar autorização sem autenticação. Por outro lado, a autorização e a autenticação confiam na auditoria, não para implementação, mas para garantir que toda exceção seja capturada para análise e irrecusabilidade.

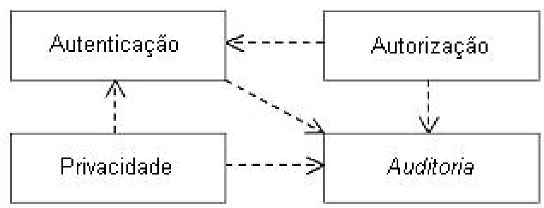

Figura 1: Dependências entre Domínios de Segurança

Esse documento continua a tratar dos planos iniciais que são comuns nesses domínios e, em seguida, descreve como esses planos iniciais, uma vez introduzidos em um modelo, podem conduzir implementações em qualquer tecnologia determinada.

Quem É Você?

Esse domínio se preocupa principalmente com a identificação de uma ou mais partes para uma comunicação ou colaboração. Podemos realmente separar isso em duas questões distintas: a noção estática de identificação e a noção dinâmica de autenticação. No exemplo da ATM, o cartão ATM é a identificação estática, emitida pelo banco, enquanto o par de cartão/PIN permite que a ATM me autentique dinamicamente como o portador da conta. É muito mais importante modelar a noção de autenticação do que de identificação; em geral, a identificação se torna muito mais vinculada aos detalhes técnicos de sua implementação do que ocorre com a autenticação. Assim, por exemplo, podemos declarar simplesmente que em uma determinada colaboração entre um conjunto de partes existe a necessidade de autenticação entre essas partes. Isso pode ser feito de forma explícita ou implícita; por exemplo, outra noção, esta de confiança, pode ser utilizada para descrever zonas confiáveis. Partes da mesma zona de confiança podem assumir que elas não precisam de autenticação entre si, ao passo que, em toda comunicação sensível além dos limites confiáveis, a autenticação é necessária.

Embora a zona de confiança seja um conceito útil, ela não atende a todas as necessidades. Zonas de confiança muitas vezes são hierárquicas, e uma comunicação com as partes externas pode ser estabelecida sem a necessidade de autenticar a parte de entrada. Uma empresa poderá se ver como uma zona de confiança isolada, no sentido de que ninguém, exceto os funcionários, pode acessar os recursos de rede dentro do firewall. Entretanto, um cenário de zona isolada como esse dará uma idéia errada se, como muitas vezes acontece, houver também uma zona de confiança separando os aplicativos de ERP (Enterprise Resource Planning) e CRM (Gerenciamento de Relação com o Cliente), porque cada um destes implementa e mantém sua própria segurança em nível do aplicativo. A empresa poderá ainda se ver como membro de uma zona de confiança externa por meio de sua associação de uma extranet comercial. A esse respeito, propomos que a confiança e a autenticação sejam planos explícitos, além de sobrepostos, que possam ser aplicados a um modelo de colaboração.

O Que Você Pode Fazer?

Esse domínio é responsável principalmente por assegurar que, uma vez conhecido quem você é, possamos restringir suas opções àquelas operações que você está autorizado a executar. Isso exige a capacidade de executar autenticação, porque precisamos saber quem você é antes de decidirmos o que você pode fazer. O acionista comercial deseja identificar aquelas funções que precisam estar seguras, no sentido de sabermos quem poderá executá-las e sabermos (através da auditoria) quando elas forem executadas. Existem é claro funções que não exigem autorização, seja porque queremos que elas estejam acessíveis a todos, ou porque, por motivos de desempenho, identificamos uma zona de confiança na qual os serviços podem assumir com segurança o direito do solicitante de acessá-los.

Essa noção de zona de confiança se torna importante quando consideramos o desempenho outra preocupação, às vezes concorrente, já que existe um custo associado à implementação da segurança, e às vezes esse custo é muito alto. Considere uma função que retorna o número de pedidos pendentes de um cliente — uma consulta muito simples em um banco de dados. Se nós solicitarmos autenticação, autorização e privacidade (veja a seguir), seremos forçados a aceitar custos significativos:

- Temos de pedir que o solicitante forneça credenciais
- Temos de verificar essas credenciais, provavelmente com um serviço remoto
- Temos de criptografar as informações retornadas

Se soubermos que o solicitante e o fornecedor são serviços do mesmo aplicativo, poderemos identificá-los como uma zona de confiança isolada, dispensar os custos e aproveitar os benefícios do aumento do desempenho. Do ponto de vista da arquitetura, é importante também compreender toda a comunicação que ocorre entre as zonas de confiança. Essa comunicação deverá ser minimizada e controlada o máximo possível, visto representar o mais provável ponto de falha (ou ataque) na implementação de segurança geral.

Em termos de implementação, há duas abordagens principais à autorização que são interessantes de se observar aqui.

- Autorização de Partes Individuais — À cada parte pode ser designado um conjunto explícito de direitos de acesso a funções (embora, para otimizar o processo, possamos concordar em tratar a ausência de um direito de acesso explícito como aprovação ou desaprovação implícita desse direito de acesso).

- Autorização através de Funções — Várias funções podem ser criadas para cada aplicativo, e os direitos de acesso designados conforme descrito anteriormente a essas funções em vez de a partes individuais. Quando cada parte é autenticada, as credenciais fornecidas para a parte incluem a função da parte, que então determina se a parte está autorizada a acessar uma determinada função.

O ponto importante a se observar aqui é que sempre queremos excluir detalhes como esses de nossos modelos comportamentais de alto nível. Nossa intenção no estágio de modelagem de alto nível é simplesmente notar, por exemplo, que uma determinada função exige autorização antes que ela possa ser executada, sem detalhar como essa autorização será feita.

O Que Você, e Outros, Podem Ver?

A preocupação do domínio de privacidade é garantir que você veja apenas as informações que está autorizado a ver, e as demais partes não vejam as informações que não estão autorizadas a ver. As empresas consomem e geram grandes quantidades de dados, que são armazenados, manipulados e transmitidos para suportar a operação dos negócios. Temos de assegurar que as informações que por natureza são confidenciais sejam protegidas e fornecer um meio de saber quem está solicitando ou fornecendo informações e serviços. Em outras palavras, precisamos saber não apenas qual serviço foi o solicitante ou o provedor, mas qual usuário final autenticado fez parte da transação através desse serviço.

Há dois planos distintamente diferentes associados ao domínio de privacidade:

- O plano de assinar uma mensagem ou um documento (possivelmente utilizando várias assinaturas) identificando os usuários finais ou os serviços que criaram a mensagem ou o documento. Esse plano, que conta com vários padrões comuns de assinaturas digitais, é utilizado em comércio B2B, do governo e ainda mais amplamente nas comunicações corporativas via e-mail. Um documento pode ter várias assinaturas, por exemplo, uma requisição de fornecimento pode ser assinada pelo originador, por seu gerente (aprovador) e finalmente pelo departamento de compras quando o pedido foi feito; todas essas assinaturas acompanham o documento.

- O plano de garantir a privacidade de uma mensagem, seja enviando-a em uma mídia segura ou criptografando ou protegendo de outra forma o conteúdo. Esse plano é provavelmente a área com a maior variedade de opções de implementação. Por exemplo, na hipótese de uma mensagem digital transferida entre serviços, podemos considerar que a própria mensagem é criptografada pelo emissor e, em seguida, enviada por um canal não seguro, que a mensagem é enviada como texto simples por transporte seguro, como HTTPS ou TLS (ambos os quais de fato fornecem criptografia como parte do serviço, mas fora do controle do emissor) ou até mesmo que a mensagem contém um identificador de um documento enviado por outros meios seguros (como impressa em papel à prova de cópia e enviada por emissário).

Outra preocupação muitas vezes citada é a integridade de dados — a necessidade de garantir que o conteúdo de uma mensagem ou documento não possa ser alterado no curso entre as várias partes da transação. Diz-se que uma mensagem ou um documento que tenha integridade de dados é à prova de falsificação. Espera-se que se uma mensagem for considerada privada, deverá ser à prova de falsificação também; entretanto, deverá ser possível indicar que uma mensagem simplesmente é à prova de falsificação sem exigir uma solução de privacidade.

Basicamente, a função do implementador é fornecer uma solução que atenda aos requisitos de privacidade de dados em comum com as diretrizes declaradas pela empresa. Por exemplo, não seria aceitável para um Web site de e-commerce utilizar um mensageiro para transmitir o número do cartão de crédito de cada cliente ao banco para verificação, ao passo que mensageiros pode ser uma excelente escolha para a entrega de documentos altamente secretos do governo.

Não Fui Eu!

Outra área de preocupação é a noção de irrecusabilidade de origem e conteúdo, um termo que você encontrará em muitos documentos EDI. Irrecusabilidade é o nome principal da noção de que em algum ponto no futuro uma das partes de uma transação possa negar ter concluído a transação. Como alternativa, uma das partes de uma transação poderá reconhecer que a transação ocorreu, mas contestar um detalhe específico dessa transação. Por exemplo, tendo adquirido 100.000 cotas de uma ação que posteriormente perdeu valor, a parte A poderá tentar alegar que a transação foi de apenas 100 cotas. Em muitos segmentos de mercado e regiões, existem regulamentos legislativos que exigem a manutenção de registros por períodos prolongados a fim de levar a juízo tais contestações.

Sob esse aspecto, a questão de auditoria que apresentamos anteriormente deverá fornecer os recursos necessários para armazenar as mensagens. Enquanto a auditoria sozinha não necessariamente satisfaz aos propósitos de irrecusabilidade, a auditoria da troca de mensagem (prova de conteúdo) com a autenticação (prova de origem) geralmente fornece o nível necessário de prova.

Aplicando as Primitivas a um Modelo

Segue um exemplo de como nossa sugestão de perfil, descrita inteiramente na seção "Detalhes de Perfil" deste documento, poderá ser utilizada para indicar planos de segurança em um modelo simples e de alto nível de uma troca de documentos entre duas partes. A tabela a seguir resume os elementos de plano utilizados no exemplo.

Plano	Comentários
auditoria	Utilizado para indicar que a documentação especificada será auditada. Alguém pode assumir que a auditoria conterá a identidade autenticada de todas as partes, bem como todos os dados comunicados entre as partes.
autenticar	Utilizado para indicar uma parte que terá de ser autenticada no escopo de uma determinada colaboração.
autorizar	Utilizado para indicar que uma comunicação entre duas partes deve garantir que o solicitante esteja autorizado a executar o pedido.
privado	Utilizado para indicar que as informações marcadas deverão ser tratadas como privadas e que todo esforço justo (considerando as implicações técnicas) deverá ser feito para garantir que os dados sejam privados (protegidos contra visualização não autorizada) e à prova de falsificação (garantia de chegada ao seu destino sem modificação).
assinado	Utilizado para indicar que as informações marcadas incluem as assinaturas digitais das partes relacionadas ao documento.
prova de falsificação	Utilizado para indicar que os dados transferidos entre as partes devem ter a garantia de chegada ao destinatário da mesma forma e com o mesmo conteúdo e significado de quando deixaram o emissor.

Plano	Comentários
confiável	Utilizado para indicar um conjunto de partes em uma colaboração que participa de uma zona de confiança explícita.

Veja a seguir o snippet de um modelo de atividade UML que demonstra a troca de ordens de compra entre um comprador e um vendedor. Observe que os elementos de plano neste exemplo são concebidos como estereótipos UML aplicados aos elementos de modelo gerais.

A figura 2 ilustra três planos de segurança para essa colaboração:

 O comprador precisa ser autenticado pelo vendedor

 A ação "Aceitar PO" exige autenticação

 O fluxo de objeto de ordem de compra deve ser assinado e em um caso deve também ser à prova de falsificação.

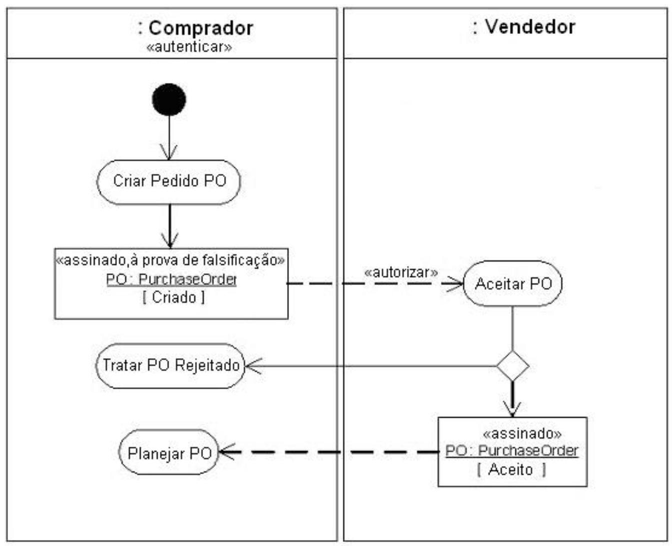

Figura 2: Planos de Segurança em Transação de Compra

É claro que o modelo não contém preocupações adicionais com tecnologia: para não mencionar serviços, terminais, interfaces, esquemas, XML e assim por diante. Esta é a visualização de alto nível e independente de plataforma que recomendamos como veículo de obtenção dos requisitos relacionados à segurança junto aos acionistas comerciais.

Vale lembrar que o perfil apresentado na seção "Detalhes de Perfil" deste documento compreende muito pouco a respeito das ferramentas ou dos métodos de modelagem que você poderá desejar implementar no desenvolvimento do modelo de plataforma neutra. Por exemplo, enquanto a Figura 2 utiliza um diagrama de atividades para modelar comportamento de negócios, a Figura 4 utiliza um diagrama de seqüência; é possível também utilizar diagramas de colaboração e até mesmo máquinas de estado para modelar o comportamento de aplicativos de negócios.

Exemplo de Mapeamento de Primitivas para Implementação

Os exemplos a seguir demonstram possíveis mapeamentos de algumas primitivas apresentadas anteriormente para determinados designs de tecnologia. Trataremos das opções reais de implementação posteriormente no documento. Novamente, suponhamos uma abordagem MDA em que modelos de alto nível são transformados em modelos específicos de implementação com o uso das transformações de modelo para modelo, que nesse caso estão incluídas nos padrões descritos a seguir.

Protocolos e Padrões

Em primeiro lugar, precisamos de um meio para expressar a implementação técnica. No UML, colaborações modeladas são utilizadas para representar padrões que então podem ser ligados a determinados elementos de modelo para expandi-los com detalhes adicionais. Em nosso caso, precisaríamos desenvolver padrões que representassem implementações específicos de tecnologia para cada plano no modelo.

A figura 3 mostra o padrão de autorização de um método que utiliza um serviço de validação confiável. Observe que o grupo de TI poderá perfeitamente ter um catálogo de padrões dentre os quais escolher para diferentes implementações ou características potencialmente diferentes, como desempenho. No exemplo, você pode ver que o padrão adota três parâmetros: o objeto solicitante, o método a ser autorizado e o serviço de validação a ser utilizado. A figura 3 demonstra também o uso de nossa anotação de zona de confiança (a caixa que contém os objetos fornecedor e validador), que visualmente chama nossa atenção para o requisito de que nosso serviço de validação seja confiado ao fornecedor do método autorizado. Isso é importante porque o conhecimento de que existe uma zona de confiança nos permite dispensar a implementação da segurança entre dois serviços que poderão interagir freqüentemente, no caso de desejarmos otimizá-los para desempenho.

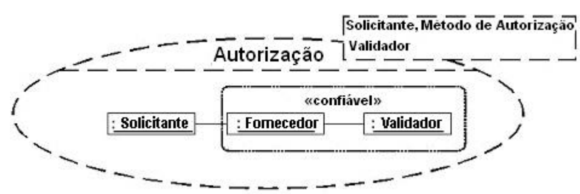

Figura 3: Padrão de Autorização

Dentro do padrão, podemos então descrever o comportamento da implementação. A figura 4 mostra como o fornecedor chama o autenticador para validar as credenciais do solicitante e, dependendo da resposta do autenticador, executa o método ou sinaliza uma exceção de autorização.

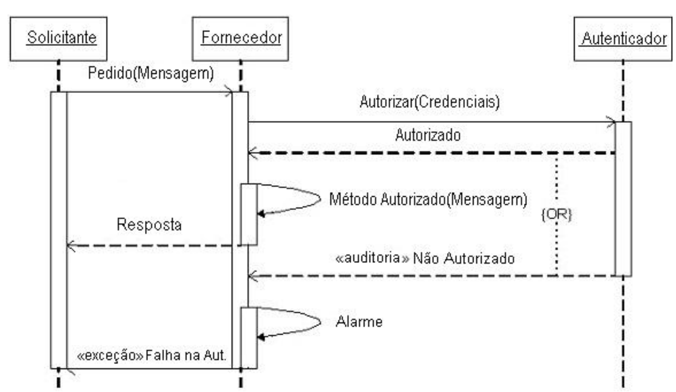

Figura 4: Comportamento de Implementação

Esse exemplo utiliza um diagrama de seqüência de mensagens UML que mostra a seqüência de eventos entre as três partes definidas na Figura 3. Observe que estereótipos adicionais foram introduzidos na Figura 4 e, particularmente, que cada resposta de mensagem do serviço de validação que indique uma autorização não aprovada será auditada.

Mencionamos anteriormente que a maioria dos planos explícitos implica em um requisito de auditoria. Ainda na Figura 4, podemos ver que a resposta NãoAutorizado do Autenticador está marcada explicitamente com o estereótipo de auditoria. A razão disso é que existe um requisito de negócios (além de todo requisito de segurança implícito) para auditar esse evento.

Observe que o estereótipo de exceção faz parte da especificação UML principal e não do nosso perfil proposto. Observe também que a restrição {OU} é utilizada para modelar em um único diagrama ambos os resultados possíveis do método de autorização.

Para conectar o padrão ao nosso exemplo de modelo de atividade, precisamos substituir os parâmetros no padrão pelos elementos do nosso modelo, conforme mostra a Figura 5. Criamos uma ligação que tem o objeto Comprador como solicitante, a ação Aceitar PO como método protegido e um serviço chamado XWSKeySvr como validador. Isso cria uma instância do padrão no modelo. Podemos então ligar o mesmo padrão a muitas outras instâncias de mensagens como "Aceitar PO" em nosso modelo de plataforma neutra.

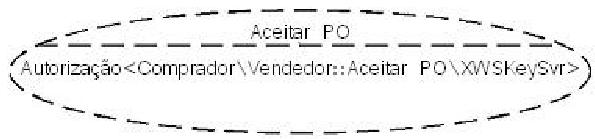

Figura 5: Ligando o Padrão ao Modelo

Essa ligação persiste no modelo e, portanto, nos permite em data posterior manipular as ligações para criar implementações alternativas (por exemplo, alterar o validador de um serviço para outro).

Opções de Implementação

Ao desenvolver os padrões de design utilizados para conduzir a implementação de planos, como autorização ou privacidade, há várias abordagens de design e soluções técnicas específicas de plataforma. Por exemplo, já discutimos a noção de separação das partes das funções na implementação de autorização. De fato, essa é uma abordagem muito comum e pode ser vista tanto no J2EE quanto no middleware Microsoft .NET. Não é o foco deste documento entrar em detalhes sobre essas decisões e padrões de design, mas esperamos que os padrões SOA específicos apareçam, sejam como variações dos padrões comuns, como o Gangof-Four[6], ou como padrões inteiramente novos específicos das restrições técnicas e da realidade de uma infra-estrutura de aplicativo orientada a serviços.

Os padrões também terão bastante influência na adoção do SOA e na forma do que exatamente o SOA se torna. Os padrões básicos, como os do W3C (World Wide Web Consortium) ou da OASIS (Organization for the Advancement of Structured Information Standards), formarão a base. Padrões de mercado, como os do RosettaNet, fornecerão o conteúdo e os processos que unirão as empresas tanto internamente quanto com seus parceiros. Entretanto, devemos tomar cuidado para não pintarmos um quadro muito prometedor. Muitos desses padrões são relativamente novos, há muitos interesses comerciais e acadêmicos representados no desenvolvimento deles e até uma organização estabelecida para administrar padrões para organizá-los (WS-I).

A tabela a seguir representa uma captura instantânea das especificações atuais relacionadas à segurança no espaço de serviços da Web. O problema com essas especificações, independentemente da Web de relacionamento entre essas especificações e entre elas e as especificações XML básicas W3C e OASIS, é que além de complexas elas estão é claro sujeitas a alteração.

XML	Sist. de mensagens	Segurança	Relacionada
XML	SOAP	Criptografia XML	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	

Acreditamos que a arquitetura orientada a serviços e a implementação de serviço da Web sejam uma grande vantagem para muitas organizações de TI nas oportunidades de integração que serão abertas. Além disso, acreditamos firmemente que as tentativas de dar nova arquitetura aos aplicativos existentes utilizando serviços da Web deverão ser cuidadosamente modeladas e plenamente compreendidas; e todas as questões importantes, tratadas separadamente em colaboração com os acionistas comerciais.

Detalhes de Perfil

Esta seção descreve uma sugestão de perfil para UML que apresenta os elementos de plano como estereótipos que podem ser aplicados aos elementos UML na captura dos requisitos dos acionistas comerciais.

estereótipo de auditoria

Metaclasses

ActivityNode, Mensagem

Descrição

Utilizado para indicar que a comunicação especificada será auditada. Alguém pode assumir que a auditoria conterá a identidade autenticada de todas as partes, bem como todos os dados sendo comunicados entre as partes.

A auditoria está implícita em qualquer comunicação estereotipada "autorizar", além de estar explícita na implementação de autenticação, bem como na manipulação de exceção para dados assinados e privados.

Na aplicação a um ActivityNode, você pode anotar ações, atividades estruturadas e nós de controle (decisões, por exemplo) em um diagrama Atividade. Na aplicação a uma Mensagem em uma interação, você pode anotar as mensagens enviadas entre os elementos de modelo representados.

Propriedades

Nenhuma.

Notação

Nenhuma anotação é necessária.

estereótipo de autenticação

Metaclasses

ActivityPartition, Linha de Segurança

Descrição

Utilizado para indicar uma parte que terá de ser autenticada no escopo de uma determinada colaboração. O estereótipo é aplicado aos elementos em um modelo comportamental que representam instâncias de uma parte em uma colaboração.

Na aplicação a um ActivityPartition (para diagramas Atividade) ou a uma Linha de Segurança (para diagramas Interação), você pode anotar os elementos de modelo representados.

Propriedades

Nenhuma.

Notação

Nenhuma anotação é necessária.

estereótipo de autorização

Metaclasses

ActivityNode, Mensagem

Descrição

Utilizado para indicar que uma comunicação entre duas partes deve garantir que o solicitante esteja autorizado a executar o pedido. Esse estereótipo é aplicado a mensagens e fluxos em modelos comportamentais para indicar que o comportamento sendo chamado está protegido por uma verificação de autenticação.

Propriedades

Nenhuma.

Notação

Nenhuma anotação é necessária.

estereótipo de privado

Metaclasses

ObjectNode, Classe

Descrição

Utilizado para indicar que os dados transferidos em uma comunicação devem ser tratados como privados e que todos os esforços justos (considerando as implicações técnicas) devem ser feitos para garantir que os dados estejam seguros e à prova de falsificação.

Não aplique os estereótipos de à prova de falsificação e privado ao mesmo elemento porque privado implica em à prova de falsificação. Ou seja, a aplicação do estereótipo de privado especifica que os dados serão privados (protegidos contra visualização não autorizada) e à prova de falsificação (garantia de chegada ao seu destino sem modificação). Em comparação, a aplicação do estereótipo de à prova de falsificação especifica que os dados terão garantia de chegada ao seu destino sem modificação, sem estar protegidos contra visualização não autorizada.

Quando aplicado a uma Classe (ou elemento UML derivado), indica que esse elemento estará assinado sempre que aparecer em um modelo comportamental. O estereótipo também pode ser aplicado a uma instância em um modelo comportamental para indicar que nesse caso específico o elemento será tratado de forma especial.

Propriedades

Nenhuma.

Notação

Nenhuma anotação é necessária.

estereótipo de assinado

Metaclasses

ObjectNode, Classe

Descrição

Utilizado para indicar que os dados transferidos em uma comunicação incluem uma noção de assinatura que identifica uma parte. Observe a importância de que o elemento não precisa ser assinado pela parte que está se comunicando e que várias assinaturas podem ser incluídas. Observe que o perfil não especifica de quem a assinatura é necessária, nem quantas assinaturas são necessárias.

Quando aplicado a uma Classe (ou elemento UML derivado), indica que esse elemento estará assinado sempre que aparecer em um modelo comportamental. O estereótipo também pode ser aplicado a uma instância em um modelo comportamental e indicar que nesse caso específico o elemento será tratado de forma especial.

Propriedades

Nenhuma.

Notação

Nenhuma anotação é necessária.

estereótipo de prova de falsificação

Metaclasses

ObjectNode, Classe

Descrição

Utilizado para indicar que os dados transferidos entre as partes devem ter garantia de chegada ao destinatário da mesma forma e com o mesmo conteúdo e significado de quando deixaram o emissor.

Não aplique os estereótipos de à prova de falsificação e privado ao mesmo elemento porque privado implica em à prova de falsificação. Ou seja, a aplicação do estereótipo de à prova de falsificação especifica que os dados terão garantia de chegada ao seu destino sem modificação, sem estar protegidos contra visualização não autorizada. Em comparação, a aplicação do estereótipo de privado especifica que os dados serão privados (protegidos contra visualização não autorizada) e à prova de falsificação (garantia de chegada ao seu destino sem modificação).

Propriedades

Nenhuma.

Notação

Nenhuma anotação é necessária.

estereótipo de confiável

Metaclasses

ConnectableElement

Descrição

Utilizado para indicar um conjunto de partes em uma colaboração que participa de uma zona de confiança explícita.

O ConnectableElement nesse caso destina-se a ser o conjunto de elementos que representa as funções em uma colaboração (conforme mostrado na figura 6).

Propriedades

Nome	Tipo	Comentários
Zona	Cadeia	O nome da zona e os participantes dela se tornam o conjunto de elementos em um determinado modelo com o mesmo nome de zona.

Notação

É útil poder indicar, graficamente, um limite de zona especificamente em um diagrama de colaboração. Um exemplo disso é mostrado na especificação do padrão Autorização. Como mostra a Figura 6, a zona de confiança é indicada por um limite pontilhado traçado em redor de seus participantes.

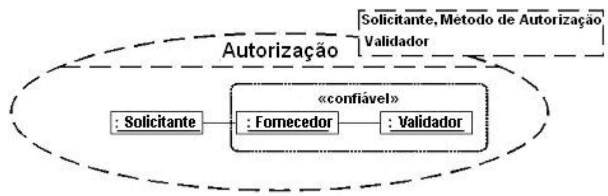

Figura 6: Indicando um Limite de Zona de Confiança

Referências

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

[2] Lopes, C.V. and Hursch, W.L., Separation of Concerns, Tech Report of College of Computer Science, Northeaster University, Boston, MA, Feb. 24, 1995.

[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., and Vlissides, J., Design Patterns, Elements of Reusable Object-Oriented Software, Addison Wesley

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produzido nos Estados Unidos da América

Março de 2005.