Soluções para o desenvolvimento de sistemas seguros

Milene Fiorio, Carlo O. Emmanoel, Paulo F. Pires e Flávia C. Delicato

SBSeg 2007

Agenda

- Motivação e visão geral de desenvolvimento de sistemas seguros
- Modelagem de Sistemas Seguros
  - Modelos de segurança
  - Padrões de Projeto de Segurança
- Construção de Sistemas seguros orientada a modelos

Cenário Atual

- Mecanismos fortes de segurança:
  - Criptigrafia, protocolos, ...
- Contudo:
  - Vários sistemas críticos em operação que não atendem os seus requisitos de segurança

Fato ocorrido em 1997

- Invasão dos computadores:
  - Depto de Defesa dos EUA
  - Sistema de controle de energia elétrica.
    - Time de hackers da NSA
      - Simularam falhas de energia e chamadas de emergência
        - Washington, D.C..
- Vários outros exemplos ...
  - Sistemas operacionais
  - Sistemas de web, ...

Causas - 1

- Projetar sistemas seguros é complexo e caro.
- Projetistas de sistemas freqüentemente não tem background em segurança.
- Segurança é um requisito freqüentemente negligênciado
  - Requisitos → ... . → Implementação

Causas - 2

- Uso "cego" dos mecanismos de segurança:
  - Ataques: Contornam mecanismos de segurança (Versus quebrar)
    - Exploram falhas existentes !

"Those who think that their problem can be solved by simply applying cryptography don`t understand cryptography and don`t understand their problem" (Lampson < > Needham).

Solução

- Segurança → Propriedade Holística
- Deve ser aplicada onde a semântica da aplicação é compreendida
- Problema "multi-nível"
  - Começar com políticas de alto nível e mapeá-las para os níveis mais baixos
- Necessidades de modelos precisos para especificação de requisitos de segurança

Cuidados

- Segurança X usabilidade (e acessibilidade)
  - ⇧ segurança → sistema inútil
  - ⇧ usabilidade → sistema inseguro
- Fator custo:
  - Usabilidade e seguraça → alto custo

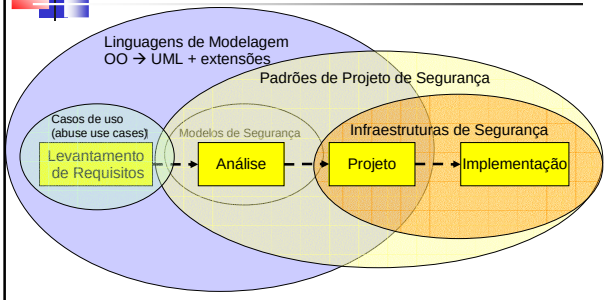
Segurança Adequada

- "A desired enterprise security state is the condition where the *protection strategies* for an organization's critical *assets* and business *processes* are commensurate with the organization's *risk appetite* and *risk tolerances*." –
  - CERT / SEI - Carnegie Mellon University - www.cert.org/governance/
  - Risk Appetite
    - Devem ser gerenciados e minimizados
  - Risk Tolerance
    - risco resudual aceitável

Solução - 1

- Considerar os requisitos de segurança em TODO o processo de desenvolvimento de sistemas
  - Planejamento
  - Análise de requisitos
  - Projeto
  - Implementação & Testes Unitários
  - Integração & Testes de Sistema
  - Operação → políticas de uso
  - Manutenção

Solução - 2



Modelagem Orientada a objetos

- Alta capacidade de modelagem conceitual
  - Regras de negócio, requisitos não funcionais (segurança), hardware, etc ..
- Alto nível de abstração
  - Detalhamento → incremental
- Modelo com extensões formais
- Linguagem padrão (OMG) → UML
  - Uso amplo !

Padrões de Projeto

- Padrões capturam soluções recorrentes e as nomeiam
  - Projeto e comunicação em um nível alto de abstração

Infraestruturas

- Sub-sistema feitos para serem estendidos
  - Voltados para domínios particulares em termos de conceitos específicos
- 'Projeto' + 'Código' → reuso + qualidade
  - implementação de certas decisões de projeto
  - conjunto de classes abstratas e suas implementações padrão
- Segurança
  - JAAS, The Microsoft® Internet Security Framework (MISF), XWS-Security framework , etc.

Agenda

- Motivação e visão geral de desenvolvimento de sistemas seguros
- Modelagem de Sistemas Seguros
  - **Modelos de segurança**
  - Padrões de Projeto de Segurança
- Construção de Sistemas seguros orientada a modelos

Conceito de Segurança

Em sistemas de informação, segurança visa **Manter as propriedades** de um sistema de forma que ele não seja alvo de possíveis **ações danosas** praticadas por **entidades não autorizadas** junto às **informações** e **recursos** nele existentes

Conceito de Segurança

- Segurança provê proteção contra:
  - Indisponibilidade
  - Vazamento da informação
  - Leitura ou modificação não-autorizadas das informações

Conceito de Segurança

- Principais Propriedades de Segurança
  - Confidencialidade
    - A informação poderá ser acessada somente por usuários autorizados
  - Integridade
    - A informação deve ser retornada na forma em que foi originalmente armazenada
  - Disponibilidade
    - A informação deve estar disponível no momento em que seja requisitada

Política de Segurança

- **Conjunto de regras** e **serviços** que visam especificar como um sistema provê os seus **recursos** mantendo as principais propriedades de segurança: **confidencialidade, integridade e disponibilidade**

Violações de Segurança

- As violações de segurança em sistemas computacionais se traduzem como sendo a **arte de burlar** de alguma forma a **política de segurança**

Violações de Segurança Conceitos

- Ameaça
  - Ação possível que, uma vez concretizada, produz efeitos indesejáveis sobre os dados ou recursos de sistema
- Vulnerabilidade
  - Falha ou característica indevida existente no sistema oriunda de falhas de concepção, implementação ou de configuração
- Ataque
  - Ameaça posta em ação

Modelos de Segurança

- Formas de descrever as políticas de segurança
- Determina o comportamento de entidades administradas pela política de segurança
- Determina as regras que definem a evolução da política
- Permitem verificações de que a política é coerente
- Servem como guia para implementações de esquemas de autorização

Modelos de Segurança

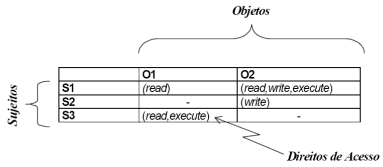
- Discricionários
- Obrigatórios
- RBAC

Modelos Discricionários

- Direitos de acesso aos recursos ou informações **especificados** para cada sujeito, **pelo proprietário** da informação ou recurso
- O proprietário decide quem tem permissão de acesso em determinado recurso e qual privilégio ele tem

Modelo Discricionário

- Modelo Matriz de Acesso
  - Listas de controle de acesso (ACLs) definem os direitos e permissões que são dados a um sujeito sobre determinado objeto



	O1	O2
S1	(read)	(read,write,execute)
S2	-	(write)
S3	(read,execute)	-

Modelos Obrigatórios

- Controle de acesso é **determinado pelo sistema** e não pelo proprietário do recurso
- Utilizado em **sistemas de múltiplos níveis** cujos dados são altamente sensíveis, como algumas informações governamentais e militares

Modelos Obrigatórios

- Prevê que os usuários, objetos e recursos do sistema sejam **rotulados**
- Um **rótulo** de um **sujeito define** o seu **nível de confiança**
- Um **rótulo** de um **objeto define** o **nível de confiança necessário** para acessá-lo
- Para acessar um determinado objeto, o sujeito deve ter um **rótulo igual ou superior** ao requisitado pelo objeto

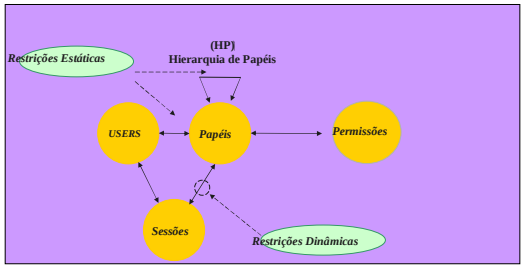
RBAC

- Controles de acesso baseados em papéis definem os **direitos e permissões baseados no papel** que determinado usuário desempenha na organização
- Permissões são conferidas aos papéis e os usuários são autorizados a exercer papéis

RBAC

- Modelo organizado em **quatro níveis** de recursos e funcionalidades crescentes
  - RBAC Básico ($RBAC_0$)
    - Prevê apenas **usuários, papéis e permissões**
  - RBAC Hierárquico ($RBAC_1$)
    - Introduz uma hierarquia entre papéis
  - RBAC com Restrições ($RBAC_2$)
    - Introduz as restrições entre papéis
    - Intenção de evitar a concentração de poder nas mãos de um único usuário
  - RBAC Simétrico ($RBAC_3$)
    - Constitui a união da hierarquia de papéis do $RBAC_1$ às restrições do $RBAC_2$

RBAC Consolidado



Agenda

- Visão geral de desenvolvimento de sistemas seguros
- Modelagem de Sistemas Seguros
  - Modelos de segurança
  - **Padrões de Projeto de Segurança**
    - **Mapeamento p/ infraestrutura de segurança**
- Construção de Sistemas seguros orientada a modelos

Modelagem de Sistemas Seguros

- Grande variedade de arcabouços de segurança disponíveis ☺
  - Ainda há noticias sobre vulnerabilidades e falhas de segurança em sistemas de software ☹
- Arcabouços se tornam rapidamente ultrapassados ☹
- *Arquitetos de software e analistas necessitam cada vez mais criar seus próprios arcabouços*

Modelagem de Sistemas Seguros

- **Grande lacuna**: soluções teóricas e o que é de fato implementado em segurança
- **Requisito de segurança**: raramente considerado nos estágios iniciais do desenvolvimento de *software*

Modelagem de Sistemas Seguros

- Necessidade:
  - Novas formas de desenvolver software seguro
    - Aplicação das teorias existentes
    - Adoção de um processo de desenvolvimento que considere os requisitos de segurança como parte integral do projeto de construção de *software*

Agenda

- Fundamentos de Segurança em Ambiente Computacional
- Modelagem de Sistemas Seguros
  - **Padrões de Projeto de Segurança**
  - Segurança com MDA

Padrões de Projeto de Segurança

- **Padrões de projeto**: introduzidos como uma forma de identificar e apresentar soluções a problemas recorrentes na programação orientada a objetos
- Muitos problemas de segurança são novos ou complicados ☹
- Muitos problemas são bem conhecidos e possuem soluções bem estabelecidas ☺
  - É melhor ensinar a pescar a dar o peixe

Padrões de Projeto de Segurança

- Os padrões de projeto descrevem soluções para problemas recorrentes no desenvolvimento de sistemas
- Soluções são desenvolvidas e conhecidas por especialistas
  - Tornam-se padrões por serem reutilizadas várias vezes em vários projetos
  - Possuem eficácia comprovada

Padrões de Projeto de Segurança

- Vantagens da utilização de padrões de projetos
  - Capturar o conhecimento e a experiência de especialistas em projeto de software.
  - Definir um vocabulário comum para a discussão de problemas e soluções de projeto
  - Facilitar a documentação e manutenção da arquitetura do software
  - Auxiliar o projeto de arquiteturas mais complexas
  - Fornecer soluções que já foram testadas e aprovadas
  - Tornar o sistema mais fácil de entender e manter

Padrões de Projeto de Segurança

- Não há um formato único e padronizado para representar padrões de projeto
- Um exemplo de formato comumente utilizado é o da gangue dos quatro que contém
  - **Nome**
    - Um nome descritivo e único que ajuda a identificar e referenciar um padrão
  - **Intenção**
    - Uma descrição do objetivo do padrão e a razão para utilizá-lo
  - **Também conhecido como**
    - Outros nomes para o padrão
  - **Motivação**
    - Um cenário contendo um problema e um contexto onde esse padrão possa ser utilizado
  - **Aplicabilidade**
    - Situações onde este padrão possa ser utilizado, o contexto do padrão
  - **Estrutura**
    - Uma representação gráfica do padrão. Diagramas de classe e diagramas de interação podem ser utilizados para este propósito

Padrões de Projeto de Segurança

- **Participantes**
  - Uma listagem das classes e objetos utilizados no padrão e seus papéis no projeto
- **Colaboração**
  - Uma descrição de como classes e objetos utilizados no padrão interagem entre si
- **Conseqüências**
  - Uma descrição dos resultados, efeitos causados pela utilização do padrão
- **Implementação**
  - Uma descrição de uma implementação do padrão
- **Código Exemplo**
  - Uma ilustração de como o padrão pode ser utilizado em uma linguagem de programação
- **Usos conhecidos**
  - Exemplos de utilizações reais do padrão

Padrões de Projeto de Segurança

- Na maioria das vezes, padrões de projeto são utilizados quando:
  - Há problema com o projeto
  - Há problema com a implementação Ao examinar um problema, é necessário procurar uma referência de padrão de que corresponda ao problema que se deseja resolver

Padrões de Projeto de Segurança

- Padrões de projeto podem parecer abstratos à primeira vista ☹
- Eles se tornam mais concretos à medida que são utilizados ☺
  - Á medida que o vocabulário dos padrões de projeto se torna conhecido, a interação se torna mais precisa e rápida com outras pessoas que utilizam este vocabulário

Padrões de Projeto de Segurança

- Padrões de projeto podem aumentar ou diminuir a capacidade de compreensão de um projeto ou de uma implementação ☹
  - É preciso saber quando e como utilizar padrões de projeto
  - Um padrão só deve ser utilizado quando a flexibilidade e os benefícios que ele oferece forem realmente necessários

Padrões de Projeto de Segurança

- **Grande lacuna**: soluções teóricas e o que é de fato implementado em segurança
- A abordagem de padrões de projeto pode e deve ser aplicada à segurança como forma de preencher essa lacuna

Padrões de Projeto de Segurança

- Vantagens de utilização de padrões de projeto de segurança
  - Novatos podem atuar como especialistas
  - Especialistas em segurança podem identificar, nomear e discutir problemas e soluções de modo mais eficiente
  - Problemas podem ser resolvidos de uma forma estruturada
  - Dependências de componentes podem ser identificadas e consideradas de forma apropriada

Padrões de Projeto de Segurança

- Padrões de segurança devem ser utilizados quando:
  - Há um problema específico em um contexto específico
  - Quando se deseja desenvolver uma arquitetura para resolver este problema
- Há vários trabalhos relacionados ao assunto de padrões de projeto de segurança

Padrões de Projeto de Segurança

- Yoder e Barcalow [YODER, BARCALOW 1997] foram uns dos primeiros a adaptar a abordagem de padrões de projeto a segurança da informação onde apresentaram os seguintes padrões
  - **Ponto de Acesso único**
    - Prover um módulo de segurança e uma forma de autenticação
  - **Ponto de verificação**
    - Organizar pontos de verificação de segurança e suas repercussões
  - **Papéis**
    - Organizar usuários com privilégios de segurança similares
  - **Sessão**
    - Localizar informação global em um ambiente multi-usuário
  - **Visão completa com erros**
    - Prover uma visão completa aos usuários, exibindo exceções quando necessário
  - **Visão limitada**
    - Permitir que os usuários visualizem somente aquilo a que eles têm acesso

Padrões de Projeto de Segurança

- Guia técnico de padrões de segurança publicado pelo Opengroup contém a definição de alguns padrões de segurança [Blakley et al 2004]
- Separado em duas categorias
  - A primeira contém padrões de segurança que facilitam a construção de sistemas que estão sempre disponíveis
  - A segunda contém padrões de segurança de autenticação e autorização

Padrões de Projeto de Segurança

- Exemplos:
  - Ponto de Restauração (*Checkpointed System*)
    - Visam estruturar um sistema de modo que seu estado possa ser recuperado e restaurado a um estado válido caso haja falha em algum componente
  - Sistema Tolerante a falha (*Comparator-Checked Fault-Tolerant System*)
    - Visa estruturar um sistema de forma que uma falha independente em um componente seja detectada rapidamente e que não cause falha no sistema inteiro

Padrões de Projeto de Segurança

- Exemplos (Cont)
  - Sistema Protegido (*Protected System*)
    - Visa estruturar um sistema de modo que todo acesso feito aos recursos seja mediado por um guardião que reforce uma política da segurança

Padrões de Projeto de Segurança

- Em [Fernandez and Pan 2001] são discutidos três padrões que correspondem aos modelos de segurança:
  - Autorização
  - Controle de Acesso Baseado em Papéis
  - Segurança Multi-nível
- Darrel M. Kienzle and Matthew C. Elder [KIENZLE, ELDER 2002] construíram um repositório de padrões de segurança com vinte e seis padrões e três mini-padroes
  - O foco desses padrões está na aplicação de segurança na web

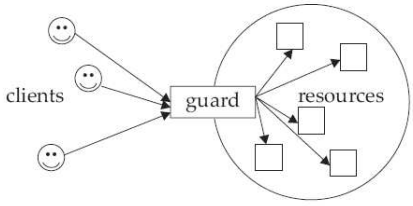
Padrões de Projeto de Segurança

- Livros:
  - Security Engineering With Patterns: Origins, Theoretical Models, and New Applications [SCHUMACHER 2003]
  - Security patterns. Integrating security and systems engineering [SCHUMACHER et al 2005]
    - Vários padrões de segurança separados por tipos
      - Padrões de identificação e autorização
      - Padrões de modelo de controle de acesso
      - Padrões de projeto de segurança para aplicações internet
  - Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management [STEEL et al 2005]
    - Escrito por um grupo da SUN e oferece um conjunto de padrões de segurança para aplicações J2EE e serviços Web

Exemplo de aplicação de padrões de projeto de segurança

- **Requisito de Segurança**: Proteger recursos contra acesso não autorizado de forma que toda requisição seja avaliada para saber se ela possui ou não permissão para acessar o recurso. Deve haver um mecanismo que:
  - É acessado a cada requisição
  - Avalia a política de segurança corretamente
  - O funcionamento correto do mecanismo não pode ser corrompido
  - Não pode haver acesso direto aos recursos

Exemplo de aplicação de padrões de projeto de segurança



Metodologia de Utilização

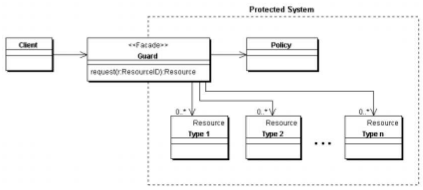
- Dedique atenção particular na aplicabilidade e nas conseqüências do padrão para ter certeza de estar utilizando o padrão correto para o seu problema
- Entenda perfeitamente as classes e os objetos no padrão e como eles se relacionam entre si
- Estude o código como forma de entender como implementar o padrão
- Os nomes dos participantes nos padrões de projeto são geralmente muito abstratos para aparecer diretamente na aplicação. Deve-se incorporar o nome do participante no nome que aparece na aplicação
- Defina as classes, declare as interfaces, estabeleça as heranças, os relacionamentos e defina as variáveis de instancia. Identifique classes existentes na aplicação que o padrão irá afetar e modifique-as de forma correta
- Defina nomes para as operações de forma específica à aplicação. Utilize as responsabilidades e colaborações associadas a cada operação como um guia
- Implemente as operações garantindo as responsabilidades e colaborações do padrão

Padrão: Sistema Protegido

- Intenção
  - Estrutura um sistema de forma que todos os acessos feitos aos recursos sejam intermediados por um guardião que valida uma política de segurança
- Aplicabilidade
  - Utilize este padrão quando o acesso a recursos deve ser consedido baseado em uma política
- Consequências
  - Recursos Isolados
  - Afrouxa o acoplamento entre a política da segurança e a implementação do recurso
  - Somente a implementação deve ser avaliada para garantir a corretude da política de segurança
  - Desempenho cai

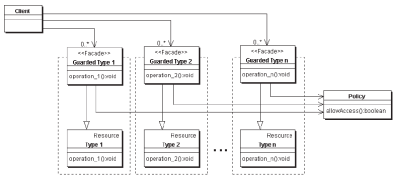
Padrão: Sistema Protegido

- Estrutura 1
  - Um guardião centrlizado avalia as requisições para todos os recursos no sistema



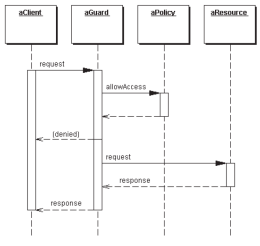
Padrão: Sistema Protegido

- Estrutura 2
  - Distribui as responsabilidades do guardião
  - Há um guardião separado para cada tipoo de recurso



Padrão: Sistema Protegido

- Colaborações



Padrão: Sistema Protegido

- Possível implementação para EJB utilizando JBoss

```
<jboss>
    <security-domain>java:/jaas/other</security-domain>
    <enterprise-beans>
      <session>
        <ejb-name>EchoBean</ejb-name>          <security-
      proxy>
            org.jboss.chap8.ex1.EchoSecurityProxy
        </security-proxy>
      </session>
    </enterprise-beans>
</jboss>
```

Padrão: Sistema Protegido

- Implementação do Guardião (EchoSecurityProxy)

```
public class EchoSecurityProxy implements SecurityProxy {
    ...
    public void invoke(Method m, Object[] args, Object bean) throws
        SecurityException {
        log.debug("invoke, m="+m); // Check for the echo method
        if (m.equals(echo)) {
          //Validate that the msg arg is not 4 letter word
          String arg = (String) args[0];
          if (arg == null || arg.length() == 4)
              throw new SecurityException("No 4 letter words");
          } // We are not responsible for doing the invoke
    }
    ...
}
```

Padrão: Sistema Protegido

- **Implementação do Cliente**

```
public class ExClient {
    public static void main(String args[]) throws Exception {
      Logger log = Logger.getLogger("ExClient");  log.info("Looking up
    EchoBean");
      InitialContext iniCtx = new InitialContext();
      Object ref = iniCtx.lookup("EchoBean");
      EchoHome home = (EchoHome) ref;
      Echo echo = home.create();
      log.info("Created Echo");
      log.info("Echo.echo('Hello') = "+echo.echo("Hello"));
      log.info("Echo.echo('Four') = "+echo.echo("Four")); }
}
```

Padrão: Sistema Protegido

- **Teste**

```
[examples]$ ant -Dchap=chap8 -Dex=1 run-example
run-example1:
    [copy] Copying 1 file to /tmp/jboss-4.0.1/server/default/deploy
    [echo] Waiting for 5 seconds for deploy...
    [java] [INFO,ExClient] Looking up EchoBean
    [java] [INFO,ExClient] Created Echo
    [java] [INFO,ExClient] Echo.echo('Hello') = Hello
    [java] Exception in thread "main" java.rmi.ServerException: RemoteException occurred in
    server thread; nested exception is:
    [java] java.rmi.AccessException: SecurityException; nested exception is: [java]
    java.lang.SecurityException: No 4 letter words
    ...
    [java] at org.jboss.chap8.ex1.ExClient.main(ExClient.java:25)
    [java] Caused by: java.rmi.AccessException: SecurityException; nested exception is:
    [java] java.lang.SecurityException: No 4 letter words
    ...
```

Agenda

- Fundamentos de Segurança em Ambiente Computacional
- Modelagem de Sistemas Seguros
  - Padrões de Projeto de Segurança
  - **Segurança com MDA**

Agenda

- Motivação e visão geral de desenvolvimento de sistemas seguros
- Modelagem de Sistemas Seguros
  - Modelos de segurança
  - Padrões de Projeto de Segurança
- **Construção de Sistemas seguros orientada a modelos**

Modelagem de Segurança

- Baseada um UML + Padrões de Projeto:
- Integra o aspecto de segurança ao ciclo de vida dos sistemas de software
- Porém ...
  - Programadores podem introduzir falhas de segurança
    - Baixa qualidade
  - Necessidade de analistas e programadores com alto grau de conhecimento de:
    - Padrões de projeto de segurança
    - Infraestruturas de segurança
    - Alto custo de desenvolvimento !

Solução

- A abordagem orientada a modelo pode auxiliar o processo de desenvolvimento de sistemas seguros
- Padrão de facto atual:
  - Model Driven Architecture (MDA)
    - OMG

O que é MDA?

"An approach to IT system specification that separates the specification of system functionality from the specification of the implementation of that functionality on a particular technology platform"

MDA specification, OMG Architecture board

- "Design once, build it on any platform"

MDA - Características

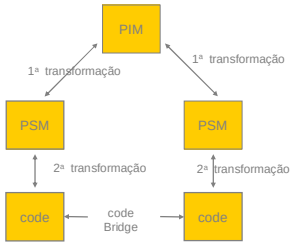
- Foco na modelagem:
  - O Sistema inteiro é representado em modelos
  - Aplicações são construídas a partir de modelos
    - Mais realista → Grande parte construída a partir de modelos
- Eleva o nível de abstração na modelagem
  - Separação de interesses
    - Verticais (e horizontais)
    - Melhor aproveitamento de equipes de desenvolvimento
- Modelagem em diferentes níveis de abstração
  - Regras (formais) de transformação
- Garante:
  - Menor esforço de implementação
  - Garantia de qualidade do produto final
    - Através da possibilidade de verificação da correção das transformações

MDA: Níveis de Modelos

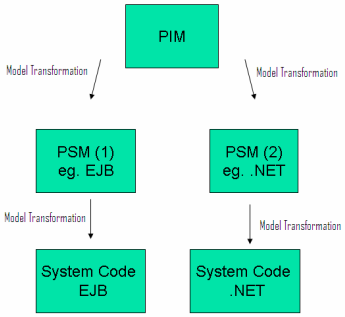
- Modelos MDA
  - Modelo Independente de Computação - CIM
    - Representação do sistema do ponto de vista independente de computação
    - Foca nos requisitos do sistema
  - Modelo Independente de Plataforma – PIM
    - É um modelo de alto nível de abstração
    - Representa as funcionalidades de um sistema
    - Não depende de plataforma
    - O PIM é transformado em um ou mais modelos PSM
  - Modelo Específico de Plataforma - PSM
    - É a complementação do Modelo Independente de Plataforma com detalhes tecnológicos específicos da plataforma de implementação do sistema

MDA: Níveis de Modelos

- Platform Independent Model (PIM)
- PIM → "código fonte"
- Platform Specific Models (PSM)
- Codigo para uma plataforma específica é gerado a partir do PSM



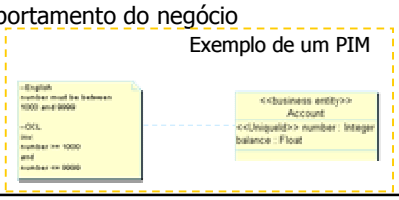
Níveis de Modelos : Exemplo



Modelo Independente de Plataforma

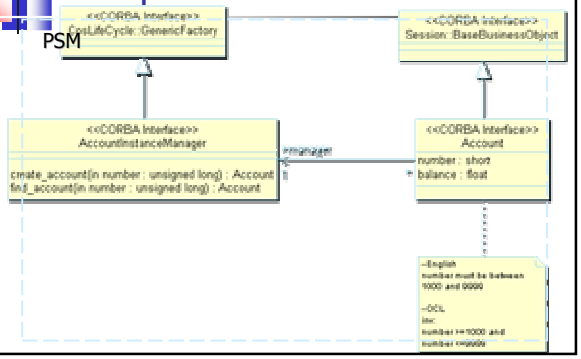
- Modelo Independente de Plataforma (PIM)
  - Expressado em UML (mecanismos estensibilidade – UML profiles)
  - Representa as funcionalidades e o comportamento do negócio

Exemplo de um PIM

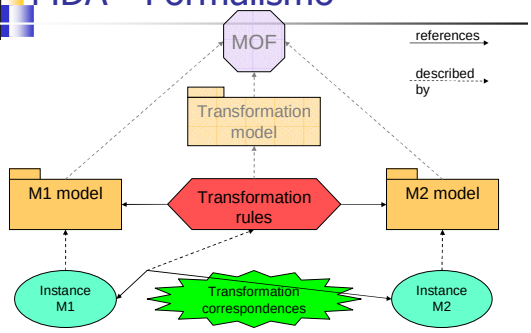


Exemplo de PSM

PSM

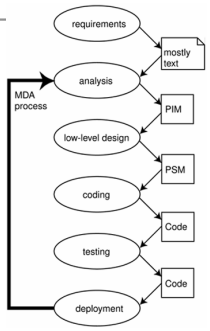


MDA - Formalismo



MDA: Processo de Desenvolvimento

- Ciclo de Vida
  - Modelo Independente de Plataforma (PIM)
  - Modelo especifico de Plataforma (PSM)
  - Código
- Automação dos passos de transformação
- Pode se aplicado com qquer processo de desenvolvimento !



Segurança com MDA

- Transformação entre modelos + Técnicas de geração automática de código
  - segurança esteja integrada a modelagem de sistemas
    - requisitos de segurança → fases do processo de desenvolvimento
  - geração automatizada de código de segurança
    - Redução de custos
    - Garantia de qualidade
  - Auxilia a criação de arcabouços de segurança flexíveis

Agenda

- Motivação e visão geral de desenvolvimento de sistemas seguros
- Modelagem de Sistemas Seguros
  - Modelos de segurança
  - Padrões de Projeto de Segurança
- Construção de Sistemas seguros orientada a modelos
  - **Abordagens**

MDA com Segurança – Trabalho 1

- **Segurança dirigida a modelo (*Model Driven Security*)** [BASIN, DOSER 2005]
- **SecureUML:** linguagem de modelagem baseada na UML
  - Integra RBAC no processo de desenvolvimento de software dirigido a modelo

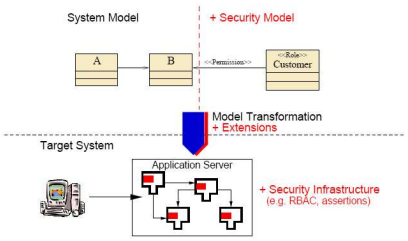
MDA com Segurança – Trabalho 1

- A informação de segurança integrada nos modelos UML é utilizada para gerar infra-estruturas de controle de acesso
- Utiliza técnicas de geração automática de código

MDA com Segurança – Trabalho 1

- O modelo do projeto + modelo de segurança = **modelo de projeto e segurança (*security design model*)**
- As políticas de segurança se referem aos elementos do modelo do sistema:
  - Componentes
  - objetos de negócio
  - Métodos
  - atributos

MDA com Segurança – Trabalho 1



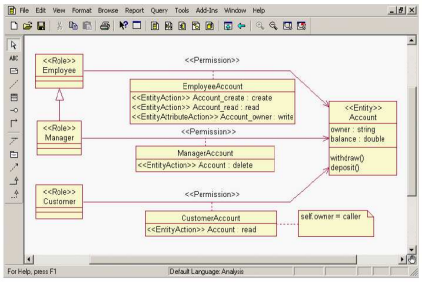
MDA com Segurança – Trabalho 1

- Foi construída uma ferramenta que gera infra-estruturas de segurança para **J2EE/EJB e .NET**
- A utilização desta ferramenta promove algumas vantagens, como por exemplo
  - Simplificação da especificação da política de segurança
  - Identificação das falhas de segurança durante o processo de desenvolvimento
  - Implementação pode ser mantida de forma consistente com a política de segurança modelada
  - Implementações podem ser migradas para novas plataformas mudando as regras de transformações utilizadas

Exemplo Utilização

- Aplicação bancária simplificada
- As entidades no exemplo são:
  - *Account* (conta), que contém os atributos *owner* (dono) e balance (*saldo*) assim como métodos *withdraw* (retirar) e *desposit* (depositar)
  - Há três papéis, que formaliza tipos diferentes de usuários:
    - *Customer* (clientes)
    - *Employee* (empregados)
    - *Manager* (gerentes)
- Política se Segurança:
  - (P1) Cada empregado pode ler informação associada com todas as contas assim como criar novas contas. Além disso, ele pode alterar o dono de uma conta.
  - (P2) Em adição às permissões dos empregados, cada gerente pode deletar contas
  - (P3) Cada cliente pode ler todas as informações associadas a suas próprias contas

MDA com Segurança – Trabalho 1



Exemplo Geração

- **O código Java abaixo mostra a pré-condição do método getBalance da entidade Account**

```
if (!(
    (
      context.isCallerInRole("Employee")
      ||
      context.isCallerInRole("Manager")
    )
    ||
    (
      context.isCallerInRole("Customer")
      &&
      context.getCallerPrincipal().getName().equals(getOwner())
    )
))
      throw new AccessControlException("Access denied");
```

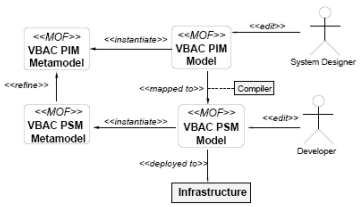
MDA com Segurança – Trabalho 2

- Em [FINK, KOCH, PAULS 2004] é apresentada uma abordagem MDA para desenvolver políticas de controle de acesso
- Os modelos são expressos como modelos MOF enriquecidos por perfis UML
  - MOF é um padrão de meta linguagem definido pela OMG a partir do qual onde outras linguagens de modelagem podem ser especificadas

MDA com Segurança – Trabalho 2

- O modelo de controle de acesso utilizado é o **VBAC (*View-Based Acces Control*)**
  - Extensão do RBAC
  - A principal característica do VBAC é ter uma visão dos direitos de acesso, que são as permissões ou negações para operações de objetos distribuídos
  - Visões são atribuídas aos principals, isto é, a sujeitos individuais ou papéis
  - Um principal possui acesso a uma operação de um objeto se ele possui uma visão do objeto com a permissão para chamar a operação
  - O principal não possui acesso se a operação está explicitamente negada em outra visão deste objeto que está disponível ao perfil ou se nenhuma permissão for achada

MDA com Segurança – Trabalho 2



MDA com Segurança – Trabalho 3

- Em [JIN 2006], é apresentado um arcabouço que utiliza a abordagem MDA juntamente com perfis UML para construir aplicações RBAC
- Gera as especificações de segurança do sistema no formato **XACML (*eXtensible Access Control Markup Language*)**

MDA com Segurança – Trabalho 3

- XACML foi desenvolvida pelo OASIS e descreve um formato para a definição de políticas de acesso em XML
- O padrão XACML define linguagens de marcação que permitem especificar políticas de segurança, requisições e respostas para decisões de controle de acesso

MDA com Segurança – Trabalho 3

- Em fevereiro de 2005, a OASIS aprovou o padrão **RBAC XACML** [RBAC XACML 2004] que define um perfil para o uso do XACML junto com os requisitos do RBAC
- Este padrão especifica quatro tipos de políticas para construir uma solução RBAC:
  - Permission <PolicySet> ou PPS
    - PPS é uma coleção de permissões associadas a um papel
  - Role <PolicySet> ou RPS
    - RPS conecta um papel aos seus PPS correspondentes que contêm as permissões atuais associadas ao papel
  - Role Assignment <Policy> ou <PolicySet>
    - Este tipo é opcional. É utilizado para responder a questão se um sujeito tem permissão para fazer parte de um papel
  - HasPrivilegeOfRole <Policy>
    - Este tipo é opcional. É utilizado para responder a questão se um sujeito possui privilégios associados ao papel

MDA com Segurança – Trabalho 3

- O processo de desenvolvimento é feito da seguinte forma
  - O desenvolvedor cria o modelo independe de plataforma (RBAC XACML PIM) baseado nos requisitos de controle de acesso da aplicação
  - Depois, o RBAC XACML PIM é transformado no modelo específico de plataforma (RBAC XACML PSM) que é utilizado para gerar os arquivos da infra-estrutura de segurança

MDA com Segurança – Trabalho 3

- Após a geração automática, o desenvolvedor implementa as partes que faltam, compila e testa o sistema
  - Neste trabalho, foi utilizado o pacote SunXACML, que foi projetado e desenvolvido pela empresa *Sun Microsystems*, é uma API que implementa a especificação XACML
  - Essa biblioteca é constituída por um conjunto de classes Java que interpretam a linguagem XACML

MDA com Segurança – Trabalho 3

- Neste trabalho, uma ferramenta de modelagem RBAC XACML foi projetada e implementada como um componente acoplável do Eclipse
- Esta ferramenta foi elaborada para demonstrar a abordagem MDA proposta
- Ela permite que o usuário crie modelos visuais para aplicações RBAC e utilize técnicas de geração de código para automatizar a construção de sistemas a partir desses modelos

MDA com Segurança – Trabalho 3

- Esta ferramenta também provê
  - um editor visual que permite o usuário visualizar e editar modelos RBAC XCAML graficamente
  - transforma e gera automaticamente arquivos de segurança no formato XACML
  - valida automaticamente o modelo para evitar erros no projeto

MDA com Segurança – Trabalho 4

- Todas as abordagens descritas até o momento, permitem a definição de políticas de controle de acesso utilizando perfis UML (estereótipos, valores etiquetados) e OCL para especificar restrições
- O PIM é modelado com requisitos de segurança
  - Sobrecarga de atribuições ao desenvolvedor
  - Expõe à falhas de segurança
  - Polui o modelo de negócio
  - Dificulta compreensão

MDA com Segurança – Trabalho 4

- Foi criado um arcabouço de segurança que é dividido em dois módulos
  - um responsável pelas **regras de segurança**, chamada de arcabouço de regras de segurança
  - um responsável pela **administração de segurança** específica de cada aplicação, chamada de arcabouço administrativo de segurança

MDA com Segurança – Trabalho 4

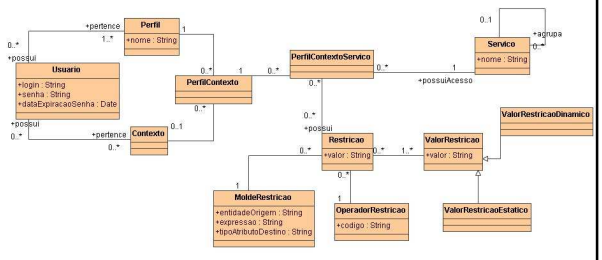
- O arcabouço de regras de segurança é o núcleo do trabalho
- Responsável pela autenticação e autorização dos artefatos da aplicação
- A segurança é aplicada implicitamente no sistema através de transformação entre modelos e codificação automática

MDA com Segurança – Trabalho 4

- **Requisitos**
  - **Ponto de Acesso único**: prover um módulo de segurança e uma forma de autenticação;
  - **Ponto de verificação**: organizar pontos de verificação de segurança e suas repercussões;
  - **Papéis**: organizar usuários com privilégios de segurança similares;
  - **Sessão**: localizar informação global em um ambiente multi-usuário; idenficar usuário logado em qualquer parte do sistema.
  - **Visão completa com erros**: prover uma visão completa aos usuários, exibindo exceções quando necessário;
  - **Visão limitada**: permitir que os usuários visualizem somente aquilo a que eles têm acesso;
  - **Sistema protegido:** estruturar o sistema de forma que todos os acessos a recursos sejam intermediados por um guardião que reforça uma política de segurança.
  - **Política:** A política de segurança deve ser aplicada em um componente discreto de um sistema de informação; assegurar-se de que as atividades onde a política de segurança é aplicada estejam sendo executadas na seqüência apropriada
  - **Descritor do sujeito:** O controle de acesso de um sujeito, isto é, uma entidade (humana ou programa) a diferentes recursos depende dos atributos desse sujeito. Descritor do sujeito provê acesso aos atributos de um sujeito e facilita gerencia e proteção desses atributos.
  - **RBAC com Contextos**
  - **Níveis de segurança**
  - **Restrições a Dados**

MDA com Segurança – Trabalho 4

- Metamodelo



MDA com Segurança – Trabalho 4

- O arcabouço administrativo de segurança representa o modelo específico de segurança da aplicação, que depende da política de controle de acesso da organização
- O arcabouço criado estende o modelo de controle de acesso RBAC com contextos
- Deixa em aberto a interpretação de usuário, papel, autorização, contexto, etc para ser especificado em modelos mais detalhados
- Um arcabouço administrativo de segurança genérico é gerado automaticamente
  - Só é necessário que a aplicação tenha que criar seu próprio modelo caso este não esteja de acordo com a política de segurança da organização

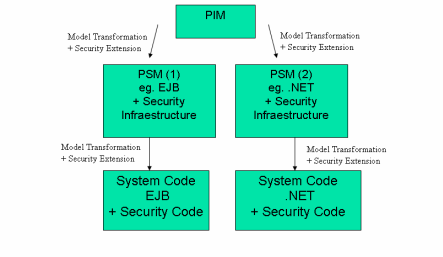
MDA com Segurança – Trabalho 4

- Processo

MDA com Segurança – Trabalho 4

- A grande mudança ocorre quando a segurança está ativada
- Durante a transformação do PIM para o PSM, os novos requisitos de segurança requerem que um novo PSM seja gerado incluindo os artefatos de segurança
- Consequentemente, a transformação de PSM para código também é alterada devido ao fato da criação de toda infra-estrutura de segurança do sistema

MDA com Segurança – Trabalho 4



MDA com Segurança – Trabalho 4

- O arcabouço foi utilizado e testado em um sistema militar brasileiro que possui uma política de segurança elevada, permitindo inúmeras vantagens em sua utilização pudessem ser percebidas.
- Ele contribui na integração de segurança durante o desenvolvimento do sistema:
  - simplificando o desenvolvimento de sistema
  - aumentando a produtividade e a qualidade das especificações de segurança
  - aumentando consideravelmente a qualidade e a manutenibilidade dos sistemas construídos
- Solução de segurança não se torna obsoleta

Exemplo geração

- Exemplo Cronus
  - Sistema de gerenciamento de projetos e processos
  - Ao ativar a segurança
    - Cadastro dos artefatos de segurança (perfis, servicos, usuarios, restrições, etc)
    - Implementação da ligação entre gerência de segurança e arcabouço de regras de segurança (se necessário)

Conclusão

- Constantemente há noticias sobre vulnerabilidades e falhas de segurança
  - Segurança é raramente considerado nos estágios iniciais do desenvolvimento de software
  - Arcabouços de segurança existentes não oferecerem um modelo arquitetural suficientemente flexível para acompanhar as necessidades do negocio

Conclusão

- São necessárias novas formas de desenvolver software seguro:
  - Baseadas na aplicação das teorias existentes
  - Baseadas na adoção de um processo de desenvolvimento que considere os requisitos de segurança como parte integral do projeto de construção de software
- Apresentamos técnicas de integração de soluções de segurança nas fases de análise e projeto de sistemas de software
  - Utilização de padrões de projeto de segurança
  - Utilização de uma abordagem orientada a modelos

Conclusão

- Apresentamos alguns trabalhos relacionados ao assunto de padrões de projeto de segurança
- Para obter os benefícios da utilização de padrões de projeto, é necessário saber quando e como utilizá-los
- Uma abordagem orientada a modelo (MDA) pode auxiliar o processo de desenvolvimento e criação de arcabouços de segurança

Conclusão

- A utilização de segurança com MDA permite que segurança esteja integrada na modelagem de sistemas
  - Modelos de projeto são combinados com modelos de segurança e técnicas de geração automática de código são utilizadas para automatizar a construção de sistemas a partir desses modelos
- As falhas de segurança podem ser identificadas mais rapidamente no processo de desenvolvimento
- Implementação é mantida consistente com a política de segurança modelada
- Implementação pode ser migrada para novas plataformas

Conclusão

- Há muitos trabalhos que integram MDA e segurança. Alguns desses trabalhos foram apresentados neste curso

Obrigado !

milenefc@gmail.com
paulo.f.pires@gmail.com
carlo@nce.ufrj.br
fdelicato@gmail.com

Referências

- BASIN, D.; DOSER, J. (2005) "Model Driven Security: from UML Models to Access Control Infrastructures. In 5th International School on Foundations of Security Analysis and Design", FOSAD.
- BLAKLEY,B.;HEALTH,C. (2004) "Security Design Patterns", Technical Guide, 2004, Doc. No. G031, ISBN: 1-931624-27-5.
- STEEL, C.; NAGAPPAN, R.; LAI, R. (2005) "Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management," Prentice Hall.
- FERNANDEZ, E.;PAM,R. (2001) "A Pattern Language for Security Models", Dept. of Computer Science & Engineering, Florida Atlantic University, 2001.
- FINK, T., KOCH, M., PAULS, K. (2004) "An MDA approach to Access Control Specifications Using MOF and UML Profiles", In Proceedings 1st International Workshop on Views On Designing Complex Architectures.
- FIORIO, M.; EMMANOEL, C. O; PIRES, P. F. (2007) "Um arcabouço de segurança baseado em transformações de modelos em MDA", Relatório técnico, Núcleo de Computação Eletrônica, UFRJ.

Referências

- JIN, X. 2006 "Applying Model Driven Architecture approach to Model Role Based Access Control System", Master of Science in System Science, University of Ottawa, Ottawa, Ontario, Canada.
- KIENZLE, D. M. ; ELDER, M. C. (2002) "Final Technical Report: Security Patterns for Web Application Development", DARPA Contract F30602-01-C-0164, disponível em http://www.scrumsecurity.org/archive/securitypatterns/dmdj_final_report.pdf, acesso em junho de 2007.
- KIENZLE, D. M.; ELDER, M. C.; TYREE, D. S.; EDWARDS-HEWITT, J. (2002) "Security Patterns Repository Version 1.0", disponível em http://www.scrumsecurity.org/archive/securitypatterns/dmdj_repository.pdf, acesso em junho de 2007.
- SCHUMACHER, M. (2003) "Security Engineering With Patterns: Origins, Theoretical Models, and New Applications", Springer Berlin, Heidelber.
- SCHUMACHER, M.; FERNANDEZ-BUGLIONI, E.; HYBERTSON, D.; BUSCHMANN, F.; SOMMERLAD, P. (2005) "Security patterns. Integrating security and systems engineering", John Wiley & Sons.
- YODER, J.; BARCALOW, J. (1997) "Architectural Patterns for Enabling Application Security", Pattern Languages of Programs, Monticello, IL.