

A Model-Driven Development for GWT-Based Rich Internet Applications with OOH4RIA

Santiago Meliá, Jaime Gómez

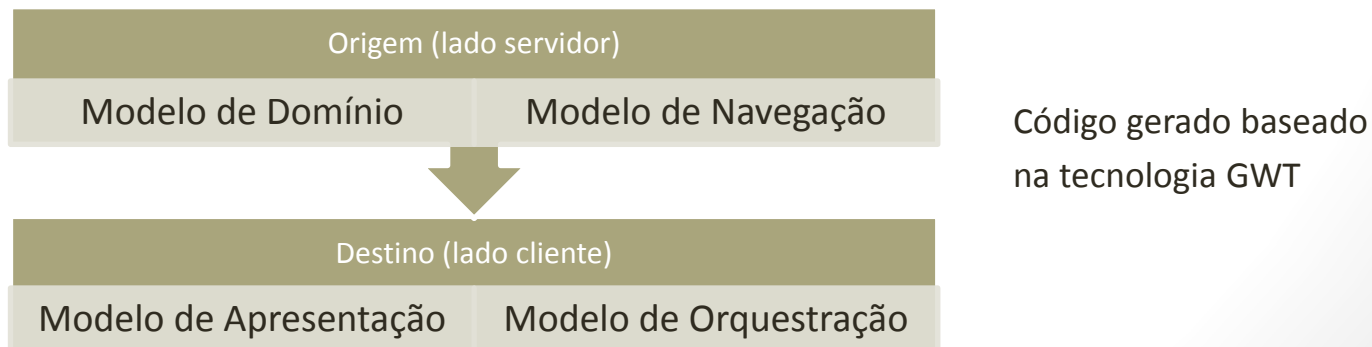
Sandy Pérez, Oscar Díaz

Introdução

- Aplicações tradicionais de cliente-servidor são estáticas;
- Advento RIA (Rich Internet Application): propõe uma interface dinâmica e interativa
 - Aumento da velocidade da atualização da página (atualização por blocos);
 - Reduz a quantidade de conteúdo trafegado;
- Por que as aplicações RIA são complexas?
 - É necessário representar interface baseada na composição de widgets;
 - É necessário definir uma coreografia baseada em eventos entre esses widgets;
 - Estabelecer uma comunicação entre as camadas de cliente e servidor;

Introdução

- RIA
 - É uma nova tecnologia;
 - Novas tecnologias demandam novos modelos;
 - Novos modelos possibilitam automação para aceleração e redução de erros;
- Proposta do artigo:
 - Um processo MDE chamado OOH4RIA (derivado do OOH) que realiza transformações M2M da seguinte forma:



O servidor RIA com OOH

- OOH (Object-Oriented Hypermedia) define:

1 – Modelo de domínio

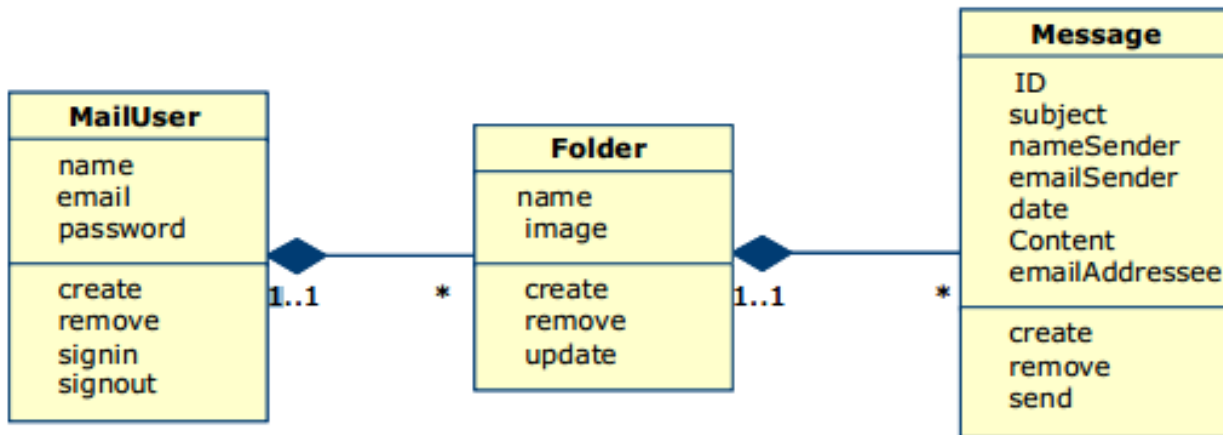
2 – Modelo de Navegação

- Baseado no MOF OMG
- Entidade Principal: Classe Navegacional (NavigationalClass)

3 – Modelo de Apresentação

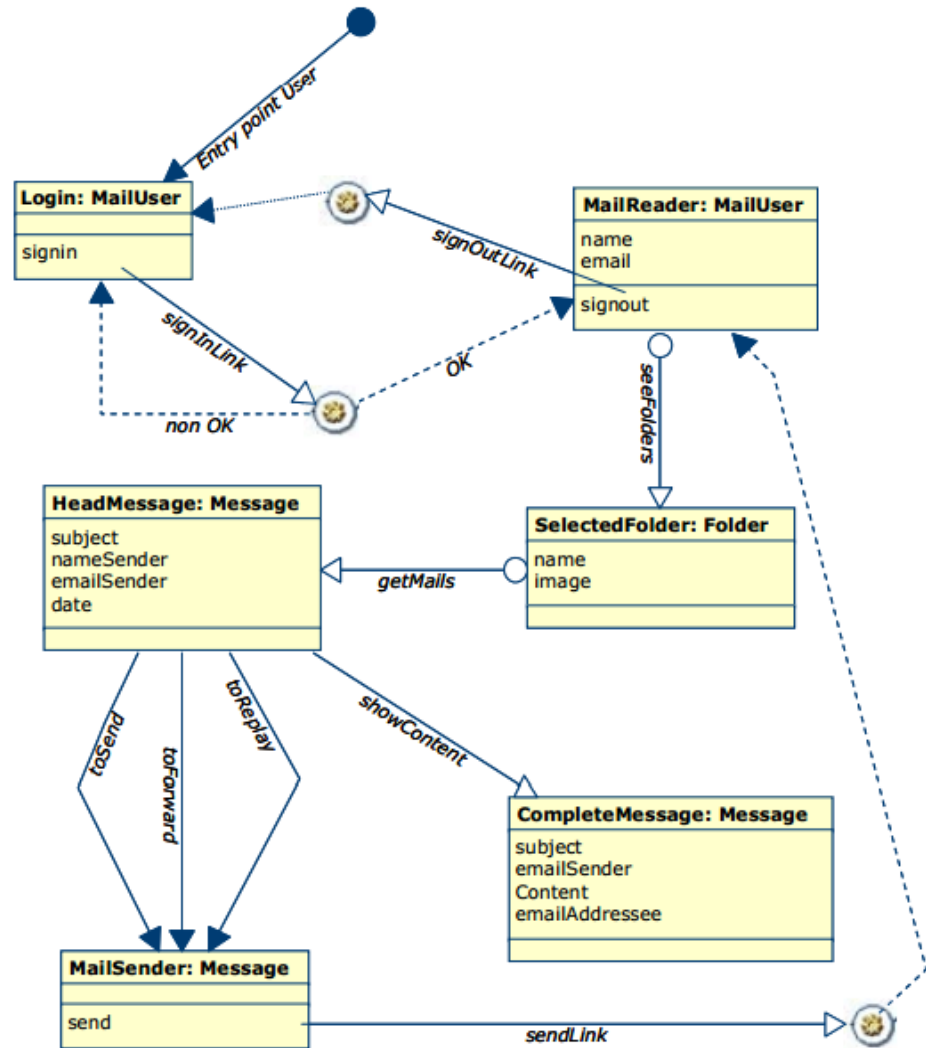
O servidor RIA com OOH

- Modelo de Domínio:



O servidor RIA com OOH

- Modelo de Navegação:
- Próximo passo: executar Transformação Nav2Pres (gera esqueletos de telas abstratas que contém apenas *screenshots* e *Panel*)

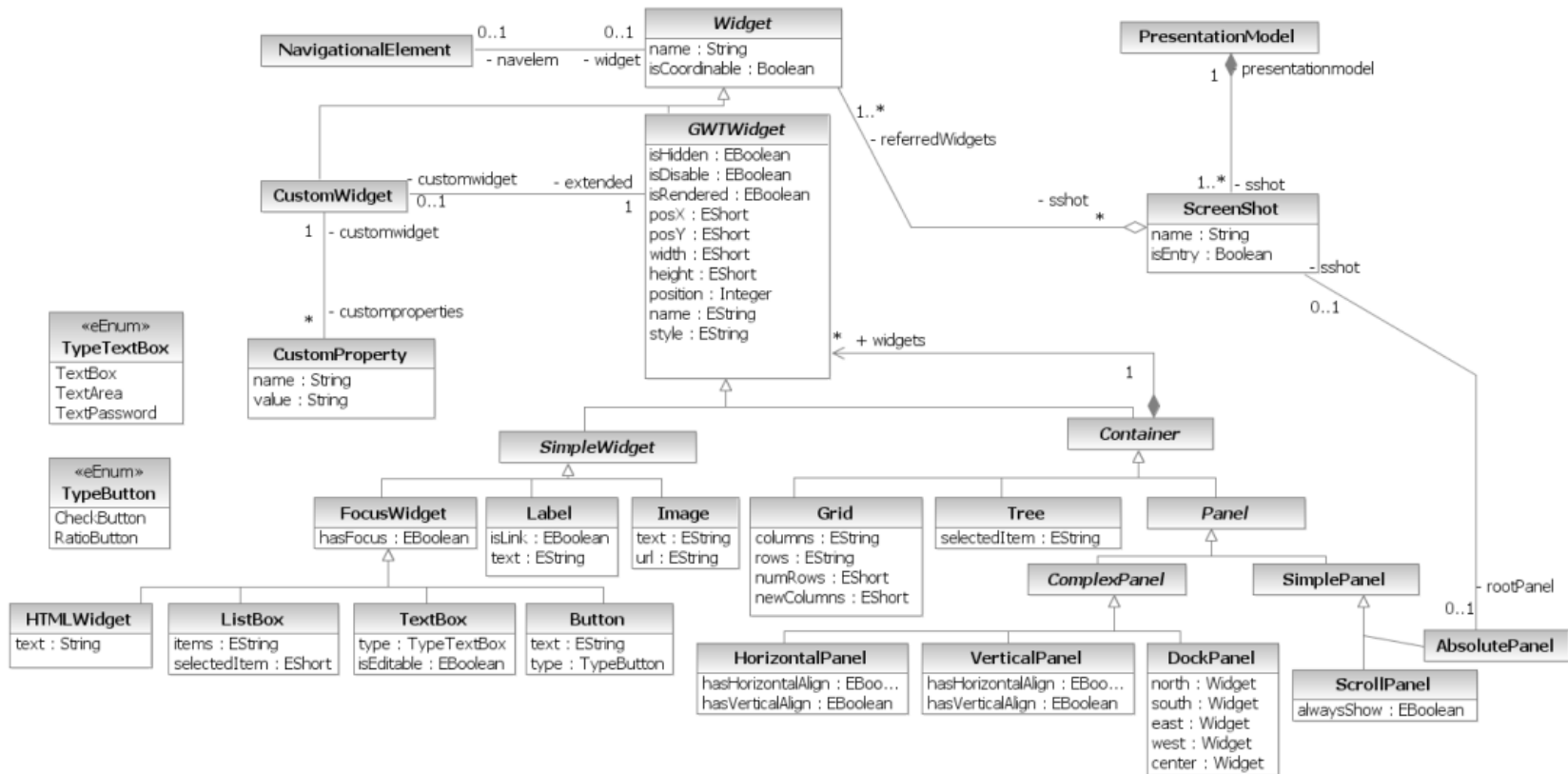


Modelo de Apresentação

- Propõe uma representação estrutural diferente
- RIA é do tipo *Simple Page Application*
- Segundo o metamodelo, todo RIA é formada por *Screenshots*
- Cada *Screenshot* tem 0 ou 1 painel e vários Widgets
- Padrão Composite para Widgets
- O modelo de apresentação é definitivamente um PSM (específico para GWT)

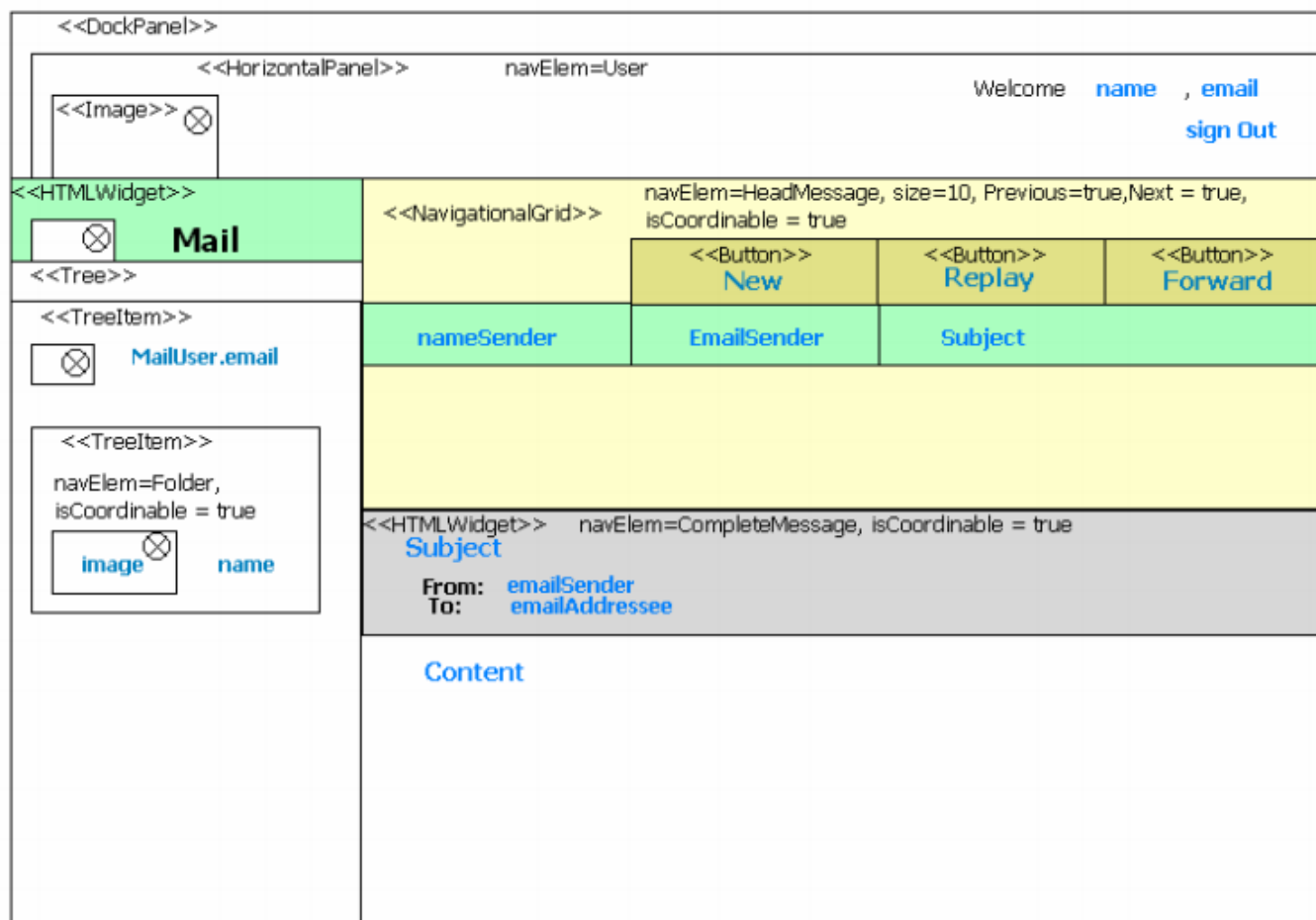
Modelo de Apresentação

- Metamodelo de apresentação Simplificado



Modelo de Apresentação

- Modelo de Apresentação da Aplicação de E-mail GWT

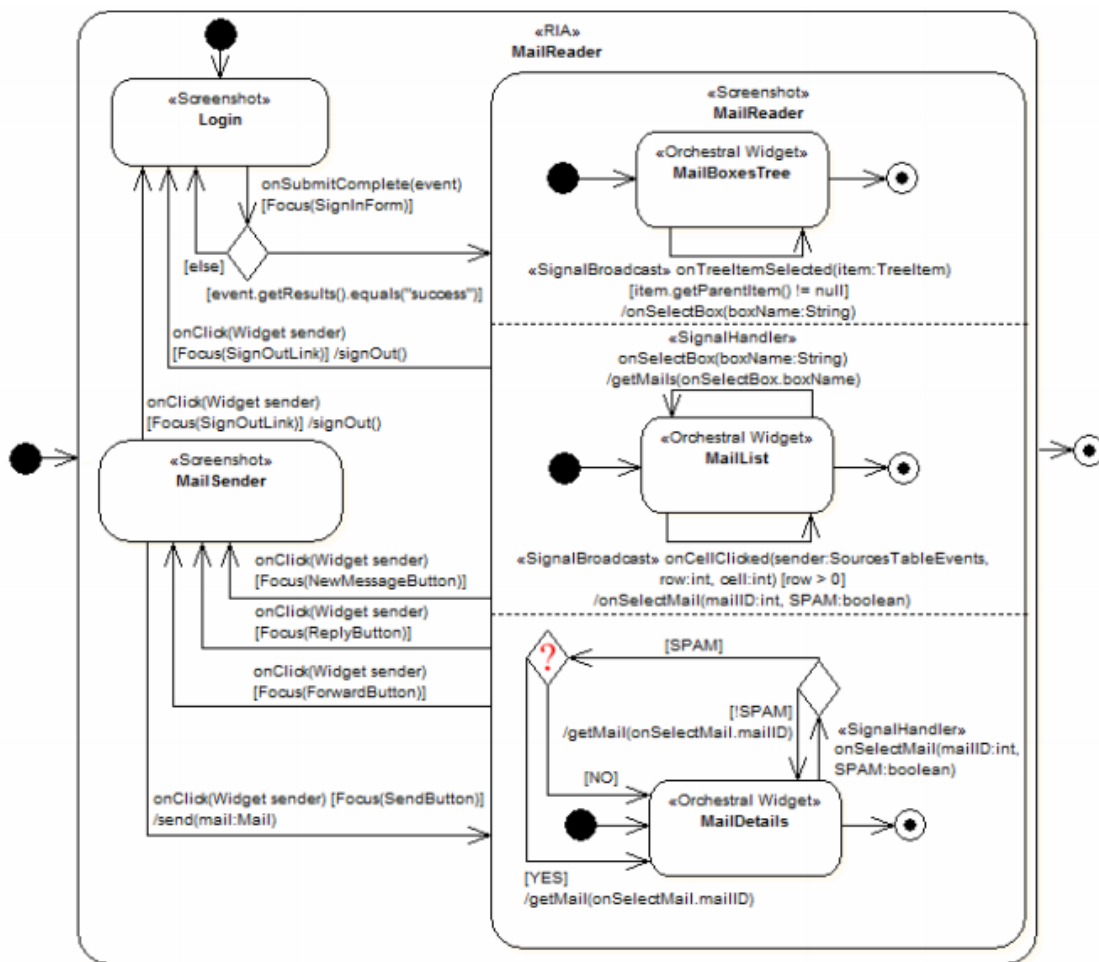


Modelo de Orquestração

- É um perfil do diagrama de estados que captura a interação dos modelos de Apresentação e Navegação dos Widgets com os Screenshots do RIA;
- Permite indicar como os widgets recebem as interações do usuário e de diferentes widgets, e como os widgets reagem enviando outros eventos para um ou mais widgets;
- *Widget Orquestral* fornece uma interação única com o usuário para alcançar uma tarefa específica (ex: mandar um e-mail);
- Define como os widgets interagem entre si;

Modelo de Orquestração

- Modelo de Orquestração da Aplicação de E-mail GWT



Modelo de Transformação

- 2 tipos diferentes:

M2M

Nav2Pres: define diferentes Screenshots para formar a apresentação da aplicação RIA

Nav&Pres2Orc: junta as informações dos modelos de navegação e apresentação e gera o esqueleto do modelo de orquestração, reduzindo o esforço de modelagem

M2T

Gera a implementação do RIA com a linguagem MOFScript

Modelo de Transformação

- Nav2Pres

```
key ScreenShot {name};
top relation NavigationModel2PresentationModel {
  nc1, nc2: String;
  checkonly domain nm: NavigationalModel {
    entryNode = n1:NavigationalNode {
      name = nc1,
      source = na:navigationalAssociation {}
    }
  };
  enforceable domain pm: PresentationModel {
    p:ScreenShot { name = nc1, isEntry = true}
  };
  where{
    if (na.samePage = true) NavNodeInSameScreenshot (na, p);
    else NavNodeInNewScreenshot (na, pm);
  }
}
```

Modelo de Transformação

- Nav&Pres2Orc

```
Key transition {name};
relation NavAssociation2Transition {
  nc1: String;
  checkonly domain n1: NavigationalAssociation {
    isSamePage = false,
    source = n1:NavigationalClass { name = nc1},
    target = n2set:NavigationalClass {}
  };
  enforceable domain t: Transition {
    source = s1:State { name = nc1},
    target = s2:PseudoState { name = nc2,
      kind = 'choice', outgoing = tset:Transition{}}
  };
  when { n2set.size() > 1 }
  where{ NavAssociation2SimpleTransition (n2set, tset)}
}
```

Modelo de Transformação

- GWT lado Servidor
 - Esta transformação converte o domínio OOH e o modelo de navegação no lado do servidor RIA, convertendo para a plataforma JAVA
- GWT lado Cliente
 - Estabelece a transformação dos modelos de apresentação e orquestração para o código GWT

Trabalhos Relacionados

Características	OOHDM	RUX + WebML	OOH4RIA
Domínio	M. Conceitual	M. De Dados(WebML)	M. Conceitual
Navegação	M. Navegação	M. Hipertexto	Diagrama de acesso de navegação
Apresentação Estrutural RIA	M. Interface	Abstrato, Concreto	M. Apresentação
Modelo		(Apresentação Espacial) e Interface Final	
Apresentação Comportamental RIA	ADV-Chart	Interface Concreta (Temporal e Interação Apresentação)	M. Orquestração
Metamodelagem	-	Parcial (WebML)	Completo
Transformação M2M	-	Geração de código	QVT
Transformação M2T	-	Geração de código	MOFScript
OMG Standards	-	UML	UML, MOF, QVT, OCL
Tool	-	WebRatio & RUX Tool	Eclipse GMF Tool (em desenvolvimento)

Conclusão

- Estende a Metodologia Web tradicional OOH através dos modelos de Domínio e Navegação e introduz transformações M2M e M2T para reduzir erros e esforços humano na geração de aplicações RIA;
- Introduz os modelos de apresentação e orquestração (principais contribuições do trabalho);
- Trabalhos em andamento
 - Geração de RIA para outros frameworks;
 - Criação de modelos arquiteturais WebSA para introdução de concepção de arquiteturas RIA no processo MDD do OOH4RIA.

Obrigado!

Laís Zagatti Pedro

Valdemar Graciano Neto