



FEUP Universidade do Porto
Faculdade de Engenharia

Geração de Interfaces Gráficas a partir de especificações VDM++

Carlos Alberto Loureiro Nunes

Orientado pela: Prof. Ana Cristina
Paiva

Conteúdo

- Introdução
- Trabalho Relacionado
- Abordagem
- Caso de Estudo
- Conclusões e Trabalho Futuro

Objectivos

“Geração Automática de Interfaces Gráficas
a partir de Especificações VDM++”

Introdução

Trabalho Relacionado

Abordagem

Caso de Estudo

Conclusões e Trabalho
Futuro

Motivação

- Dispensar o uso de Interpretadores
- Protótipo de Interface Evolutivo

Desenvolvimento de Interfaces Gráficas

- “*Toolkits*” Gráficos
- Ferramentas Interactivas (“*WYSIWYG*”)
- Linguagens de Anotação de Interfaces
- Ferramentas baseadas em linguagens formais
- “*Property Models*”
- Restrições
- Técnicas automáticas baseadas em modelos

Geração Automática de Código

- Compiladores
 - Análise Léxical, Análise Sintactica, Análise Semántica, Geração de Código
 - Representação Intermédia



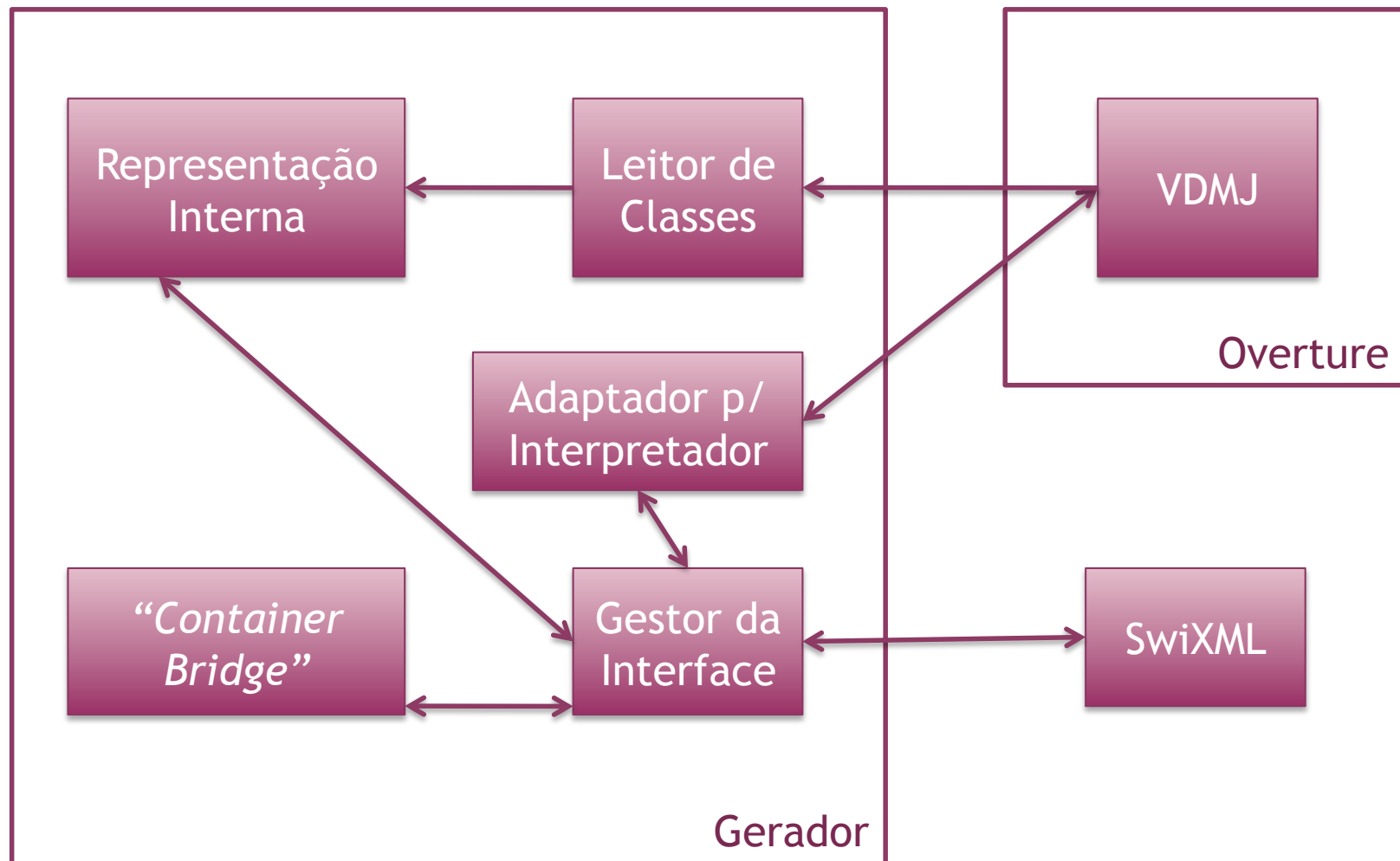
Pontos a Reter

- Técnicas/Ferramentas de desenvolvimento gráfico focam em assistir o programador, não em remover o programador;
- São necessárias técnicas de compilação para extrair a informação necessária de uma especificação VDM++.

Características

- Extracção de Informação Relevante da Especificação VDM++
- Definição da Interface (Regras de Construção)
- Geração da Interface
- Ligação da Interface com o Modelo

Arquitectura



Regras de Geração

Modo de Geração		Descrição
Sem Anotações		Cada classe VDM++ é usada como base de uma janela. Os métodos dão origem, a caixas de entrada de dados para argumentos, um botão para invocação, e um elemento de texto para o retorno.
Com Anotações	--@nowindow	Classes com esta anotação são ignoradas no processo de geração.
	--@press	O método deve ser usado no processo de geração, usando as mesmas regras que o modo sem anotações.
	--@check=<valor>	O método é usado para recolher informação. São gerados dois elementos, um que exhibe o texto <valor>, e um segundo que exhibe o retorno do método.

Introdução

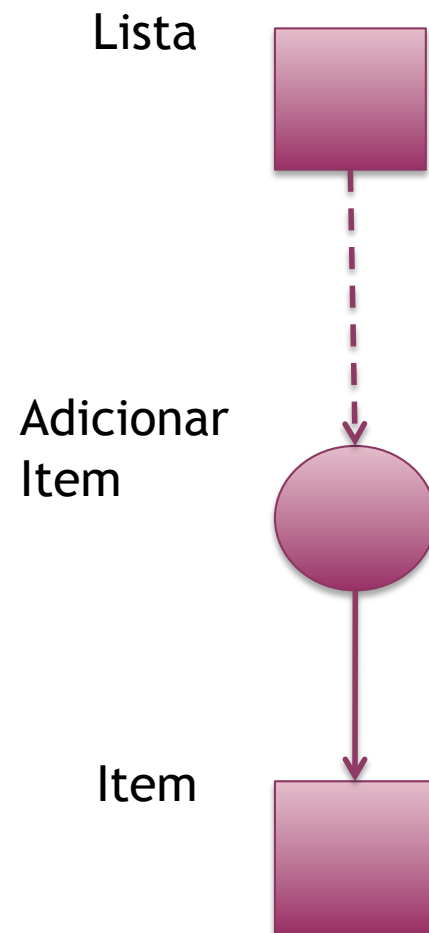
Trabalho Relacionado

Abordagem

Caso de Estudo

Conclusões e Trabalho
Futuro

Abordagem - Dependências



Introdução

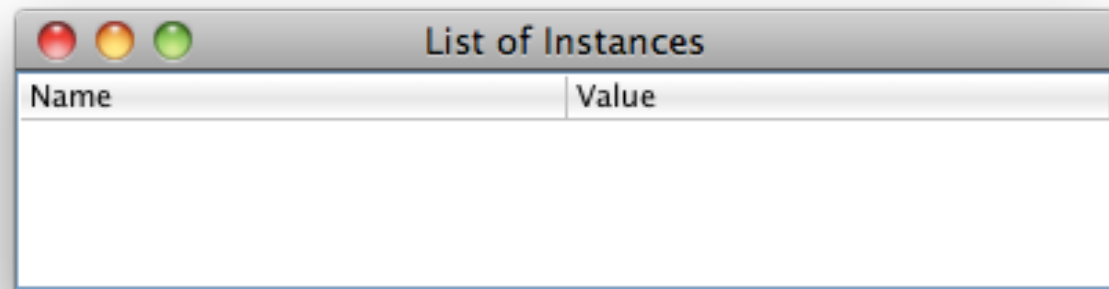
Trabalho Relacionado

Abordagem

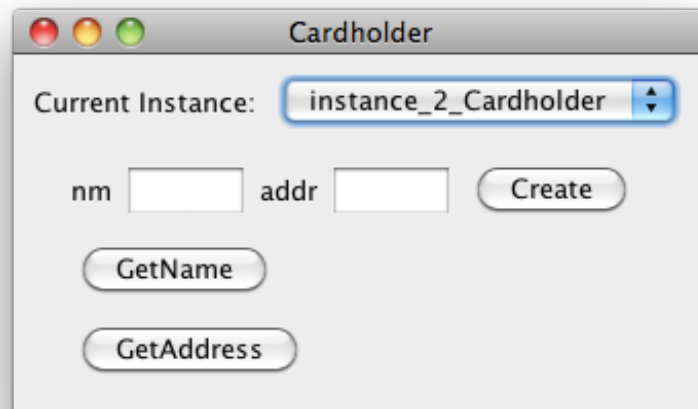
Caso de Estudo

Conclusões e Trabalho
Futuro

Sistema Multibanco



Sistema Multibanco



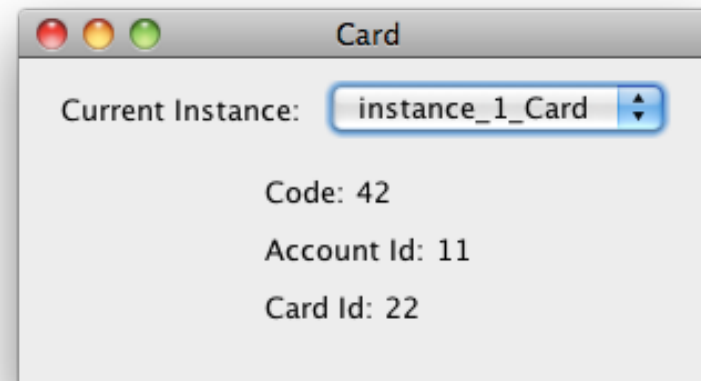
Cardholder

Current Instance: instance_2_Cardholder

nm addr Create

GetName

GetAddress



Card

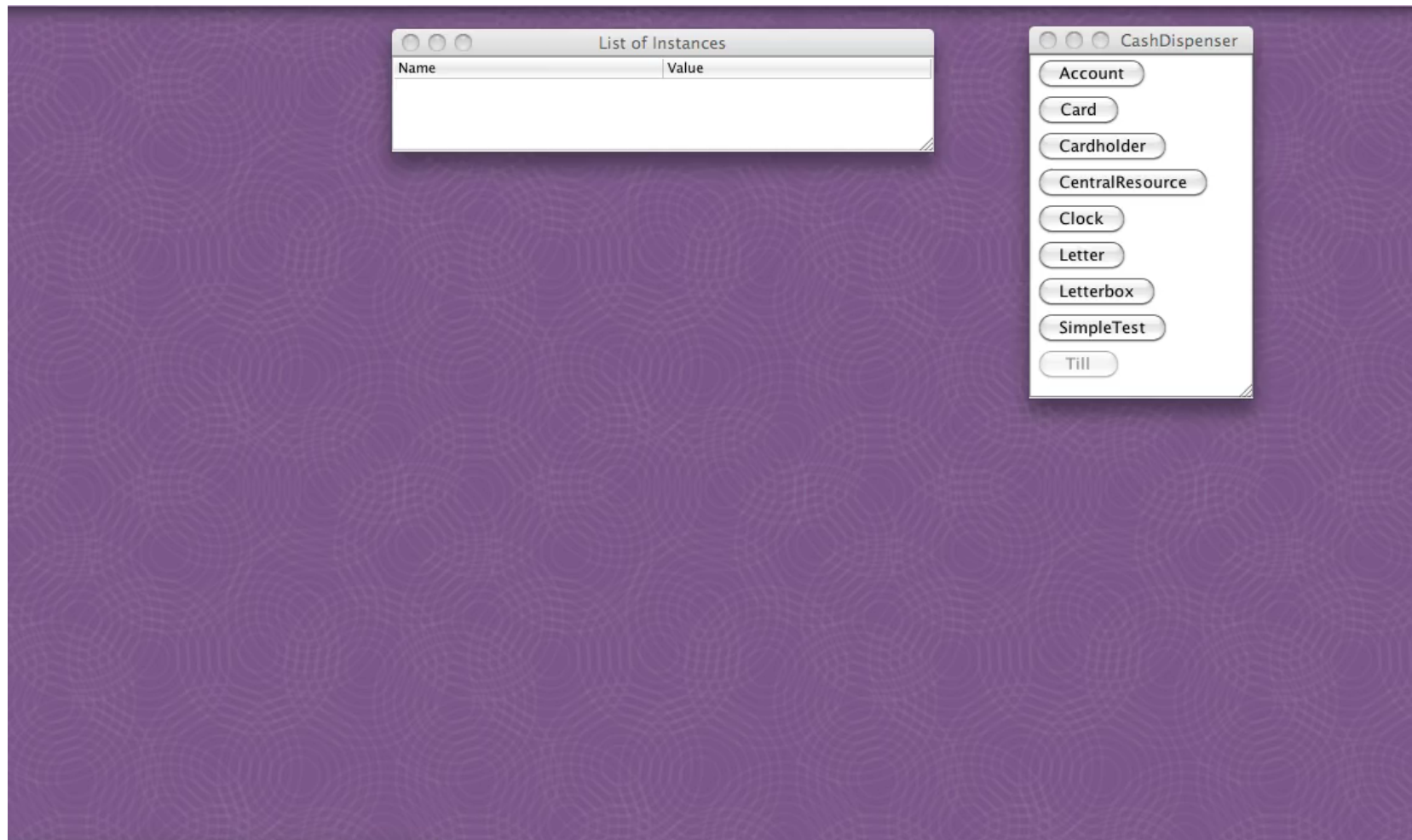
Current Instance: instance_1_Card

Code: 42

Account Id: 11

Card Id: 22

Interacção



Introdução

Trabalho Relacionado

Abordagem

Caso de Estudo

Conclusões e Trabalho
Futuro

Caso de Estudo

The screenshot displays the 'CentralResource' application window. At the top, it shows the 'Current Instance' as 'instance_1_CentralResource'. Below this, there are two dropdown menus labeled 'c' and 'I', followed by an 'AddLetterbox' button. The main interface is divided into two sections: 'Accounts' and 'Cards'. The 'Accounts' section includes a 'New Account' form with an 'accId' input field and an 'acc' dropdown menu, an 'AddAccount' button, and a note stating 'Note: The account id is only for the Central Resource.' Below this, there are input fields for 'accountId' and 'cardId', with buttons for 'GetBalance' and 'PostStatement'. The 'Cards' section contains input fields for 'accountId', 'cardId', and 'amount', with buttons for 'Withdrawal', 'IsLegalCard', 'ResetNumberOfTries', 'IncrNumberOfTries', 'NumberOfTriesExceeded', and 'AddIllegalCard'. The bottom of the 'Cards' section has a red background with the 'AddIllegalCard' button.

Resultados

- É gerada uma interface funcional para a especificação VDM++;
- A interface gerada é pouco sofisticada;
- Podem ser introduzidas mais anotações, mas estas poriam em causa o objectivo do trabalho;
- A habilitação/desabilitação de elementos verifica a existência de argumentos, mas não a sua validade.

Conclusões

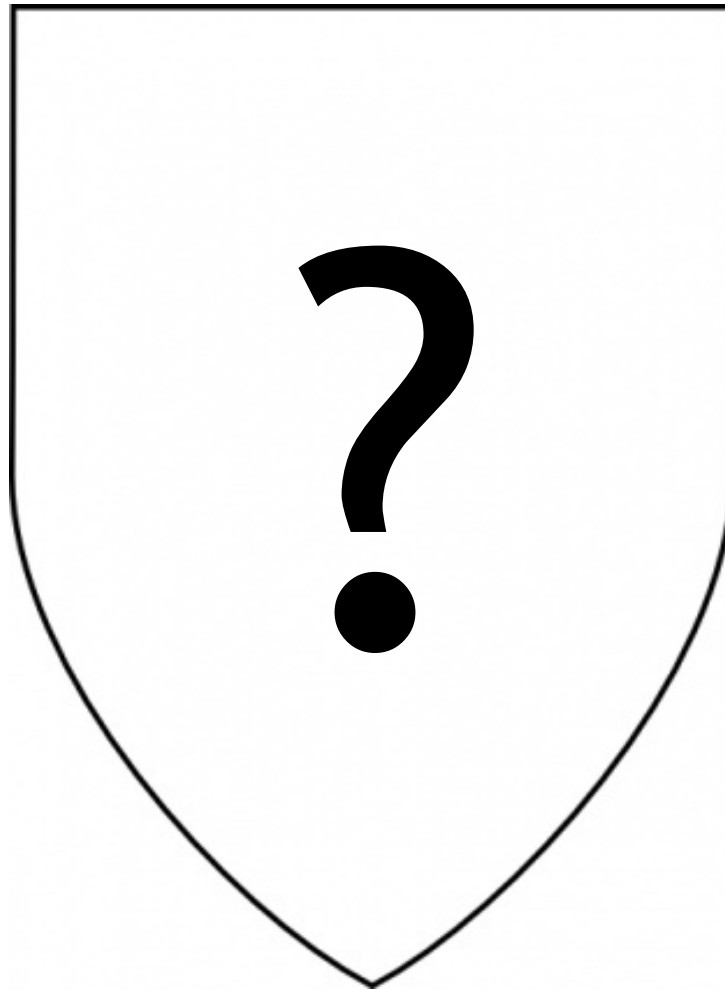
- A abordagem é capaz de gerar interfaces funcionais a partir de especificações VDM++.
 - Seguindo a gramática da linguagem;
 - Sem requerer a participação activa do utilizador.
- A adição de informação suplementar através de anotações, permite uma interface mais ajustada à especificação.
- A divisão da investigação sobre definição de interface e atribuição de funcionalidade, permitiria um maior foco nos problemas específicos de cada área.

Trabalho Futuro

- Escolha de diferentes padrões de interface.
- Tomar partido das pré-condições existentes numa especificação VDM++.
- Ligar a interface gerada ao código já gerado por ferramentas VDM existentes.
- Tornar o leitor de classes dependente da AST do projecto Overture.
- Usar uma análise mais profunda da especificação para inferir que métodos podem ser anotados com “--@check”.

- ICSEA 2011 - International Conference on Software Engineering Advances, 23-28 de Outubro, Barcelona, Espanha.
- Overture Project - Incorporação da Funcionalidade na plataforma e Interesse no documento.

Perguntas



Métodos Formais

