

Ferramenta para conversão de interfaces gráficas desenvolvidas em Delphi para a biblioteca GTK+

Acadêmico: Josimar Zimmermann

Orientadora: Joyce Martins

Roteiro

- ▶ Introdução
- ▶ Objetivos
- ▶ Fundamentação teórica
- ▶ Desenvolvimento
- ▶ Conclusão
- ▶ Extensões



Introdução

- ▶ Aplicativos construídos em Delphi exclusivamente para a plataforma Windows
- ▶ Advento das plataformas alternativas ao Windows
- ▶ Necessidade de portabilidade e construção de interface gráficas
- ▶ Tempo e mão de obra para tornar aplicativos portáteis



Objetivos do trabalho

- ▶ Usar a biblioteca GTK+ como solução portátil na construção de GUI
- ▶ Migrar GUI construídas no Delphi para um formato legível à GTK+
- ▶ Permitir a utilização da mesma GUI através de diferentes linguagens de programação



Fundamentação teórica

▶ Conceitos:

- ▶ Definição de padrões com XML
- ▶ GUI
- ▶ Desenvolvimento de GUI no Delphi
- ▶ GTK+
- ▶ Glade

▶ Trabalhos correlatos

- ▶ DelphiToWeb
- ▶ Extensão da ferramenta DelphiToJava-II
- ▶ ScriptCase



Definição de padrões com XML

- ▶ Formato padronizado
- ▶ Portável
- ▶ Adaptável
- ▶ Suporte para processamento pelas linguagens de programação
- ▶ Estrutura hierárquica



GUI

- ▶ Do inglês *Graphical User Interface*
- ▶ Utiliza a experiência do usuário
 - ▶ Imita objetos conhecidos
 - ▶ Utiliza padrões existentes
- ▶ Bibliotecas para construção de GUI
 - ▶ *Visual Component Library* (VCL)
 - ▶ Qt
 - ▶ GTK+



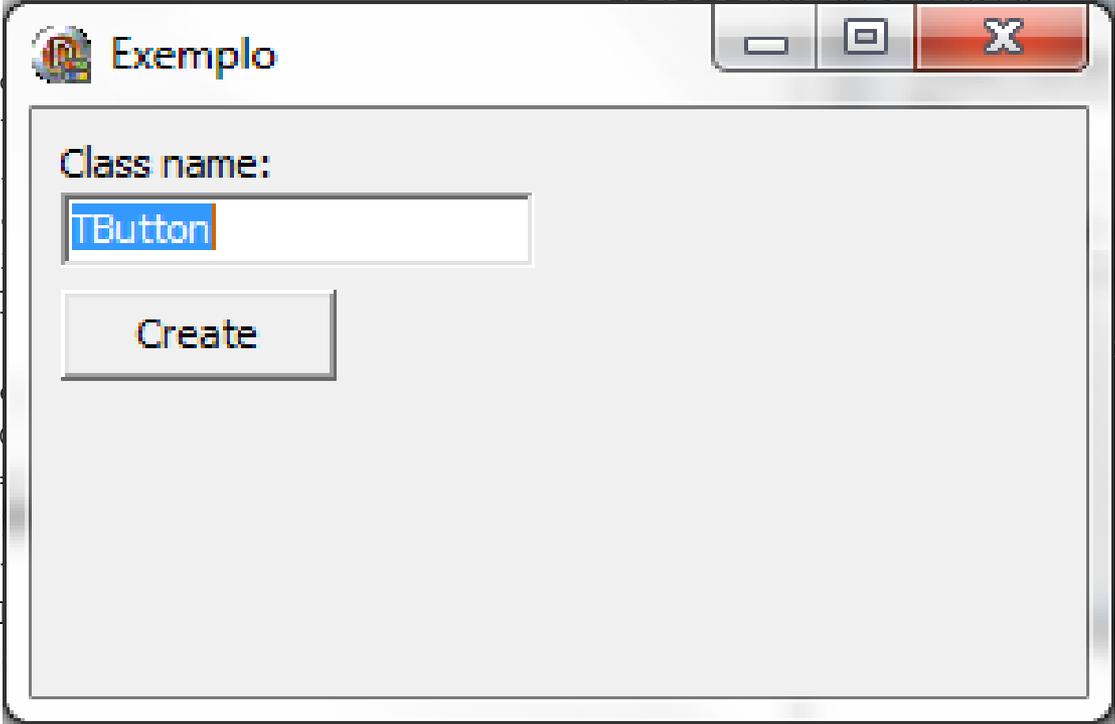
Construção de GUI no Delphi

- ▶ Armazena informações da GUI separadamente
- ▶ Utiliza um formato próprio
 - ▶ Hierarquia de componentes
 - ▶ Propriedades
 - ▶ Manipuladores de eventos



Construção de GUI no Delphi

```
object Form1: TForm1
  Left = 0
  Top = 0
  Caption = 'Class name:'
  ClientHeight = 0
  ClientWidth = 0
  Color = clNone
  Font.Charset = ANSI_CHARSET
  Font.Color = clBlack
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  Position = poDefault
  PixelsPerInch = 96
  TextHeight = 13
  object Edit1: TEdit
    Left = 100
    Top = 80
    Width = 58
    Height = 13
  end
end
```

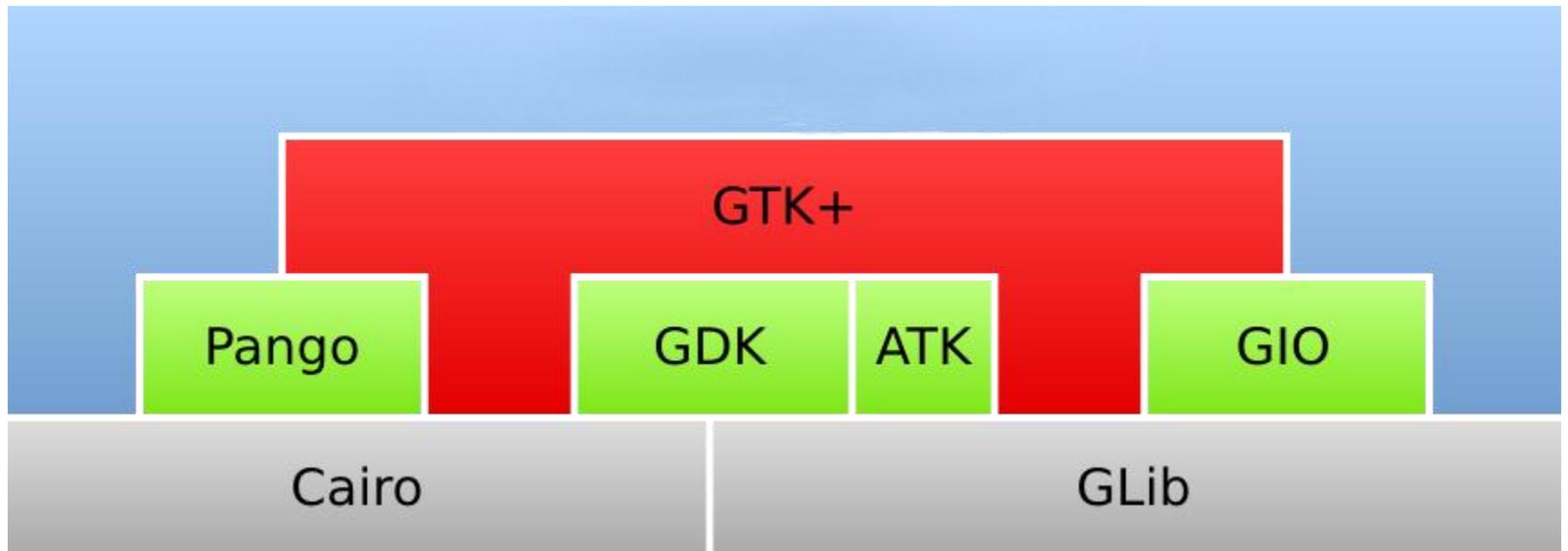


GTK+

- ▶ Acrônimo para *GIMP ToolKit*
- ▶ Criada em 1997 na Universidade da Califórnia
- ▶ Extensivamente utilizada no Linux
 - ▶ GNOME
- ▶ Expandida para suportar outros sistemas
 - ▶ Microsoft Windows
 - ▶ Mac OS X
 - ▶ Outros



GTK+



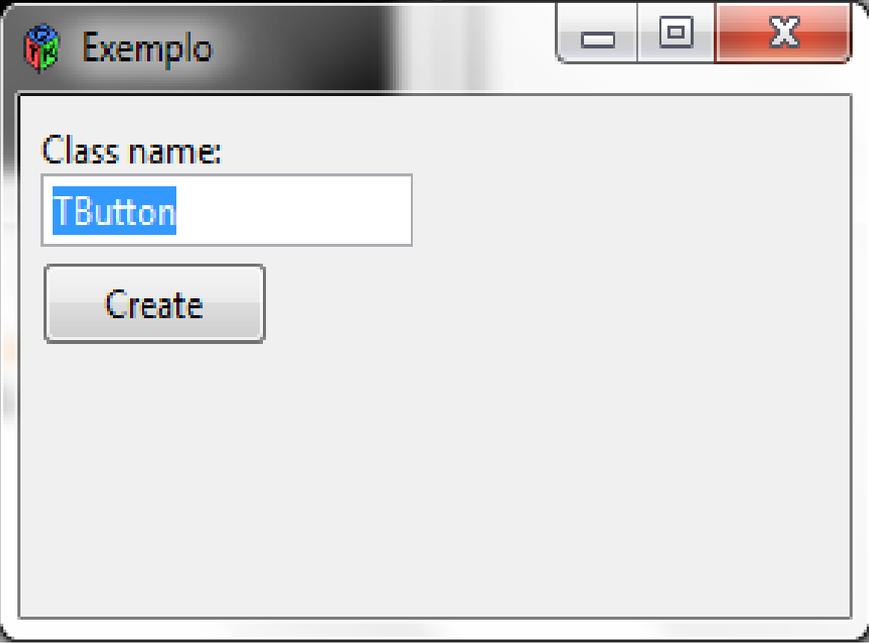
Glade

- ▶ Construtor de GUI para a GTK+
- ▶ Fornece suporte a `Libglade`
- ▶ Fornece o recurso *drag-and-drop*
- ▶ Salva as definições da interface em XML
- ▶ Construção da GUI à parte da lógica do aplicativo



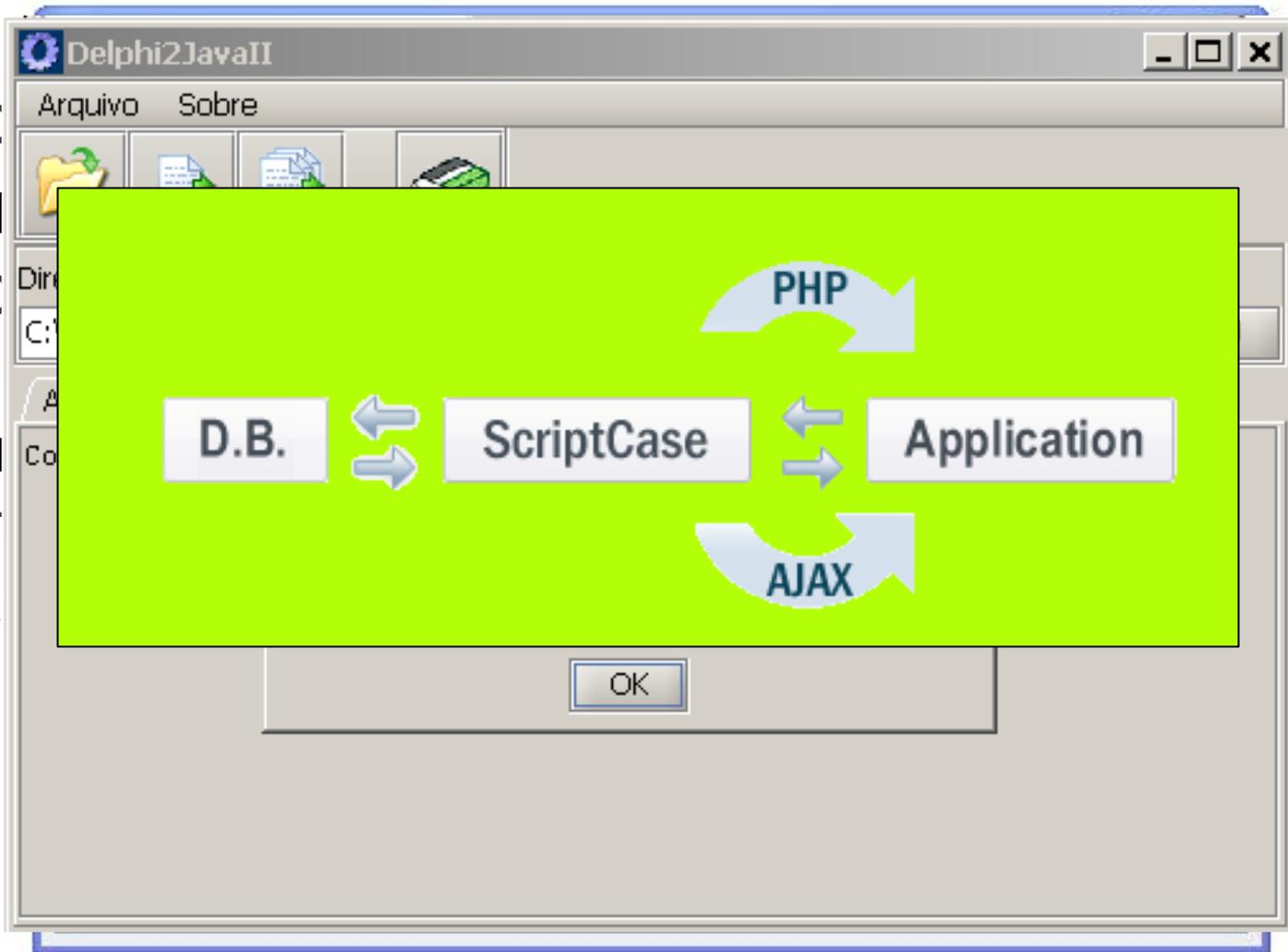
Glade

```
<?xml version="1.0"?>
<glade-interface>
  <!-- interface-requires gtk+ 2.16 -->
  <!-- interface-naming-policy project-wide -->
  <widget class="TButton"
    <property name="label" value="Create" />
    <property name="width" value="100" />
    <property name="height" value="30" />
    <signal name="clicked" handler="on_clicked" />
    <child>
      <widget class="GtkWindow"
        <property name="title" value="Exemplo" />
        <property name="width" value="200" />
        <property name="height" value="100" />
        <child>
          <widget class="TButton"
            <property name="label" value="Create" />
            <property name="width" value="100" />
            <property name="height" value="30" />
            <signal name="clicked" handler="on_clicked" />
            <child>
              <widget class="GtkLabel"
                <property name="label" value="Class name:" />
                <property name="width" value="100" />
                <property name="height" value="20" />
                <child>
                  <entry text="TButton" />
                </child>
              </widget>
            </child>
          </widget>
        </child>
      </widget>
    </child>
  </widget>
</glade-interface>
```



Trabalhos correlatos

- ▶ De
- ▶ C
- ▶ Ext
- ▶ C
- ▶ d
- ▶ Sci
- ▶ F
- ▶ a



Desenvolvimento

- ▶ Requisitos
- ▶ Especificação
- ▶ Implementação
- ▶ Operacionalidade



Requisitos

▶ Requisitos funcionais

- ▶ Permitir a seleção dos diretórios de:
 - ▶ Mapas;
 - ▶ Entrada;
 - ▶ Saída
- ▶ Realizar a conversão de `.dfm` para `.glade`
- ▶ Exibir o resultado da conversão:
 - ▶ Em execução;
 - ▶ No Glade



Requisitos

- ▶ **Requisitos não funcionais**
 - ▶ Codificar para Pascal/Object Pascal para o compilador Free Pascal
 - ▶ Implementar na IDE Lazarus 0.9.28.2
 - ▶ Desenhar a interface com auxílio do Glade
 - ▶ Utilizar analisadores léxico, sintático e semântico
 - ▶ Gerar código Pascal/Object Pascal para exibição do resultado



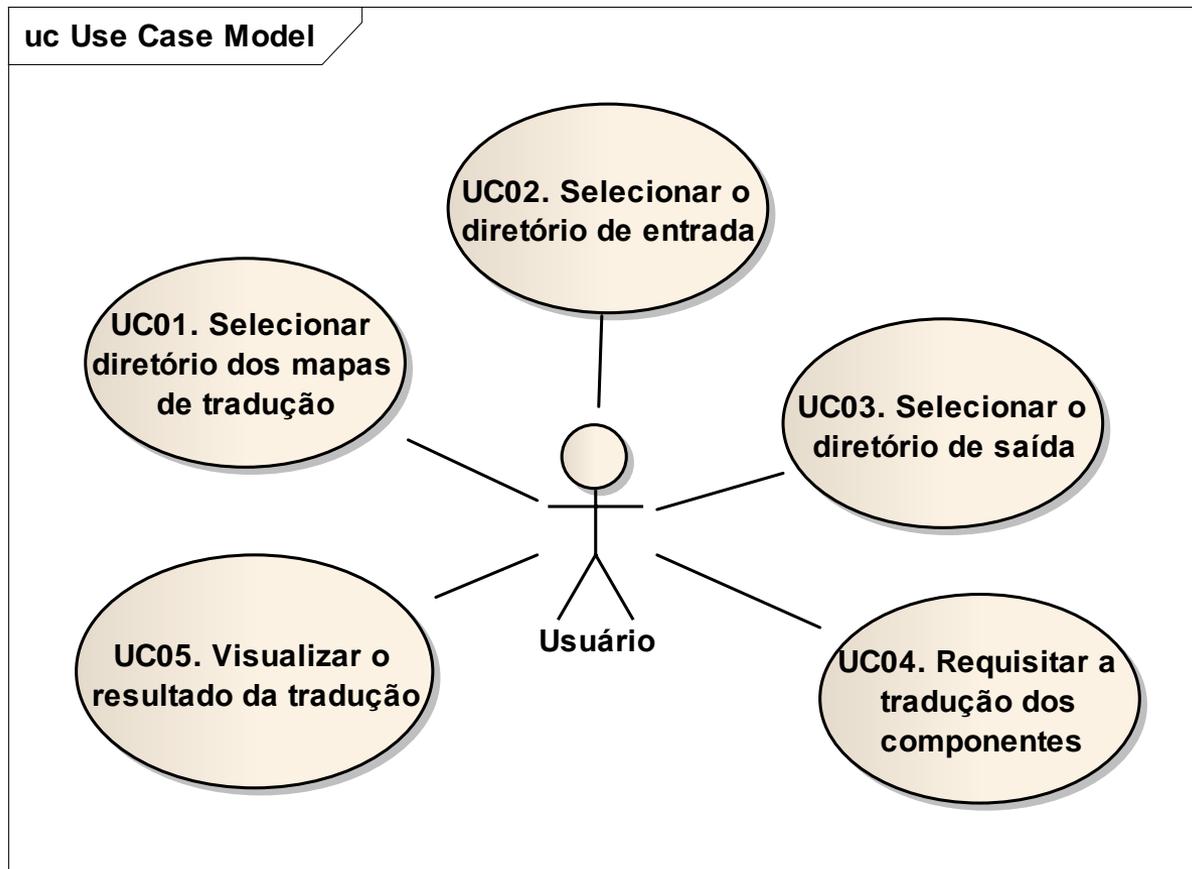
Especificação

- ▶ Ferramentas utilizadas
 - ▶ Enterprise Architect
 - ▶ gVim
- ▶ Artefatos gerados
 - ▶ Diagramas de casos de uso
 - ▶ Diagramas de pacotes
 - ▶ Diagramas de classes
 - ▶ Mapas XML para auxiliar a tradução



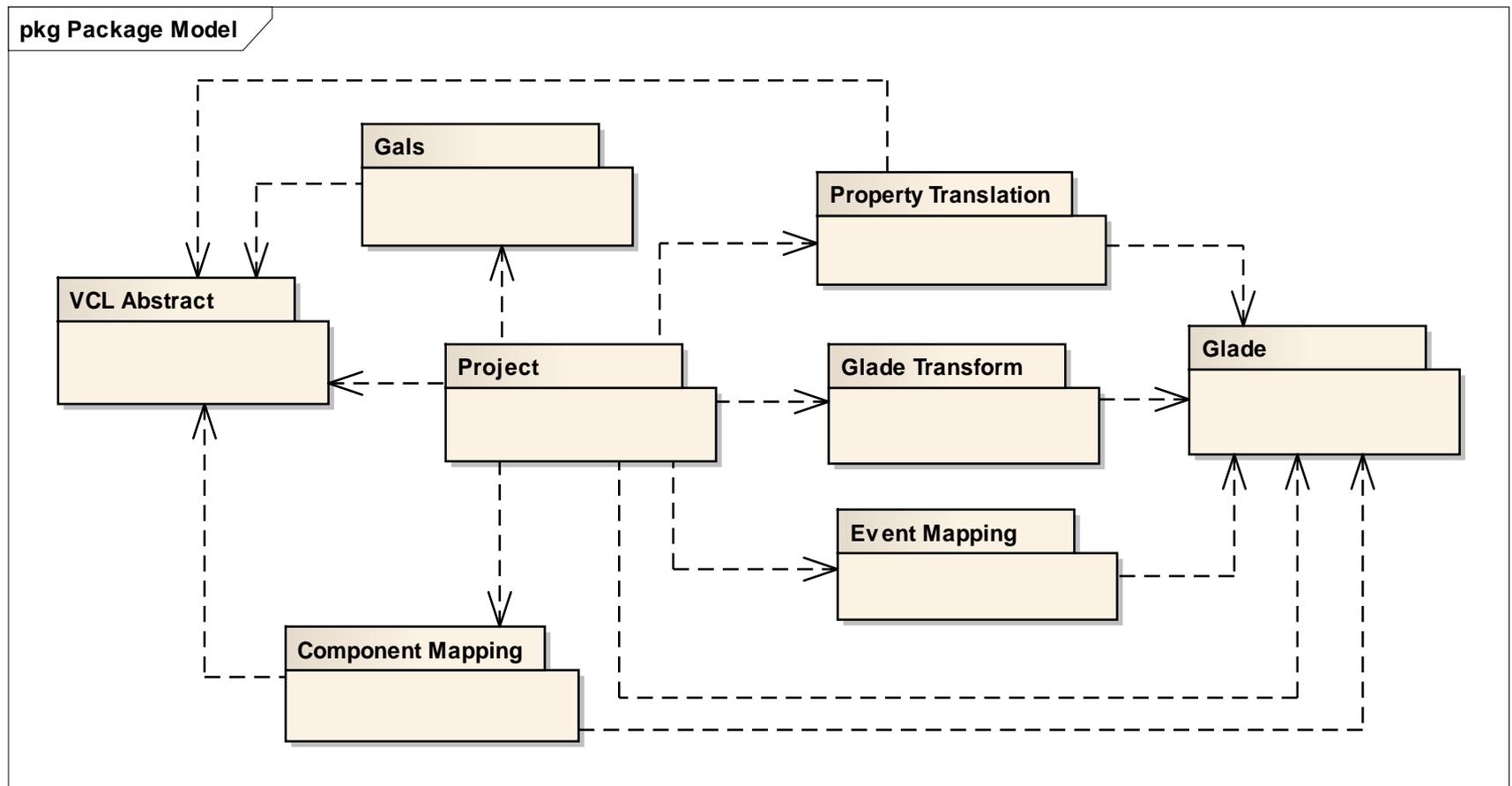
Especificação

► Casos de uso



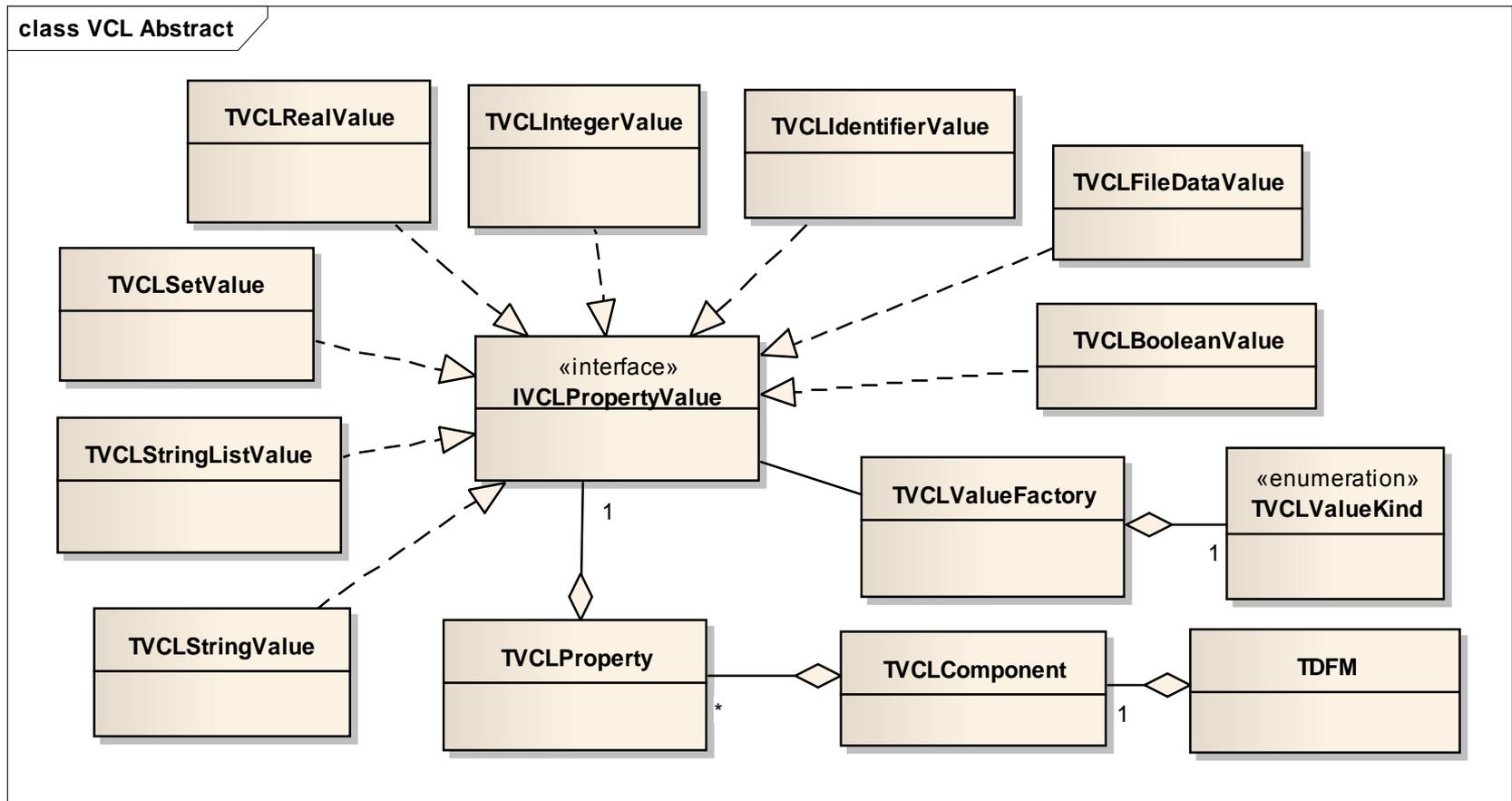
Especificação

► Diagrama de pacotes



Especificação

▶ Diagrama de classes do pacote VCL Abstract



Especificação

- ▶ Mapeamento da tradução
 - ▶ Componentes
 - ▶ Propriedades
 - ▶ Eventos
- ▶ Utilização do padrão XML
 - ▶ Hierarquias
 - ▶ Portabilidade
 - ▶ Flexibilidade



Especificação

- ▶ Mapa de componentes
- ▶ Tradução categorizada
 - ▶ Simples
 - ▶ Condicionada a propriedades
 - ▶ Especial
- ▶ Classe GTK resultante
- ▶ Propriedades requeridas



Especificação

► Componente de tradução simples

```
<?xml version="1.0" encoding="UTF-8"?>
<delphitogtk>

  <component class="TForm" translation="simple">
    <widget class="GtkWindow" />
  </component>

  <component class="TLabel" translation="simple">
    <widget class="GtkLabel" />
  </component>

</delphitogtk>
```



Especificação

► Componente de tradução condicionada

```
<component class="TScrollbar" translation="propertied">
  <widgets>
    <widget class="GtkHScrollbar">
      <properties>
        <property name="Kind" type="Identifier" check="equality"
          value="sbHorizontal" />
      </properties>
    </widget>
    <widget class="GtkVScrollbar">
      <properties>
        <property name="Kind" type="Identifier" check="equality"
          value="sbVertical" />
      </properties>
    </widget>
    <widget class="GtkHScrollbar" />
  </widgets>
</component>
```

Especificação

- ▶ Componente de tradução especial
- ▶ O tradutor é uma classe compilada

```
<component class="TPanel" translation="special"  
  translator="TPanelTranslator" />
```



Especificação

▶ Mapa de propriedades

```
<class name="TProgressBar">
  <groups>
    <group translator="Alignment">
      <properties>
        <property name="Position" required="true" type="Integer"
          default="0" />
        <property name="Step" required="true" type="Integer"
          default="10" />
        <property name="Min" required="true" type="Integer"
          default="0" />
        <property name="Max" required="true" type="Integer"
          default="100" />
      </properties>
    </group>
  </groups>
</class>
```



Especificação

▶ Mapa de eventos

```
<component class="TComboBox">
  <event vcl="OnSelect" gtk="changed">
    <arguments>
      <argument name="Widget" type="PGtkComboBox" />
      <argument name="UserData" type="GPointer" />
    </arguments>
  </event>
</component>
```



Implementação

▶ Ferramentas utilizadas

- ▶ Lazarus IDE, versão 0.9.28.2
- ▶ Biblioteca GTK+ *runtime* para Windows
- ▶ Glade, versão 3.6.7
- ▶ GALS

▶ Recursos e técnicas utilizadas

- ▶ *Generics* (disponibilizado pelo Free Pascal)
- ▶ POO



Implementação

- ▶ Carregamento dos mapas de tradução
- ▶ **Classe** `TComponentMap` carrega o arquivo `components.xml`
- ▶ **Classe** `TPropertyMap` carrega o arquivo `properties.xml`
- ▶ **Classe** `TEventMap` carrega o arquivo `events.xml`



Implementação

- ▶ Interpretação do arquivo `.dfm`
- ▶ Utilização dos analisadores léxico, sintático e semântico
- ▶ Analisador semântico alimenta o pacote `VCL Abstract`
 - ▶ Instancia a classe `TDFM`
 - ▶ Adiciona os componentes
 - ▶ Adiciona as propriedades



Implementação

- ▶ **Classe** `TDFMTranslator` realiza a tradução do objeto `TDFM`
- ▶ Utiliza os mapas para auxiliar na tradução
- ▶ Processo de tradução
 - ▶ Componentes
 - ▶ Propriedades
 - ▶ Eventos
- ▶ A tradução resulta um objeto `TGladeInterface`
- ▶ Transforma `TGladeInterface` em XML
- ▶ Gera código Pascal para carregar a GUI



Operacionalidade

- ▶ **Usuário informa a ferramenta...**
 - ▶ Diretório de localização dos mapas
 - ▶ Diretório de localização do arquivo `.dfm`
 - ▶ Diretório de saída do arquivo traduzido
- ▶ **Usuário solicita a tradução**
- ▶ **Ferramenta informa o resultado da tradução**



Operacionalidade

- ▶ Após traduzida a interface o usuário pode...
 - ▶ Visualizar a tradução no Glade
 - ▶ Visualizar a tradução em execução
 - ▶ Visualizar código fonte que carrega a interface



Operacionalidade

▶ Exemplo de tradução



A screenshot of a software window titled "Exemplo". The window contains three dropdown menus labeled "País:", "Planeta:", and "Transporte". All three dropdown menus are currently empty.

A screenshot of the same software window "Exemplo". The "País:" dropdown menu is now filled with "Moçambique". The "Planeta:" dropdown menu is filled with "Júpiter". The "Transporte:" dropdown menu is empty.

Resultados e Discussão

- ▶ Desenvolvida com software livre
- ▶ Desenvolvida para fornecer portabilidade
 - ▶ Free Pascal
 - ▶ XML
 - ▶ GTK+
- ▶ Permite a adição de tradutores para novos componentes



Resultados e Discussão

► Quadro comparativo

característica	DelphiToWeb	Extensão da ferramenta Delphi2Java-II	ScriptCase	DelphiToGTK+
linguagem de programação	Java	Java	PHP e Javascript	Pascal/Object Pascal
oferece flexibilidade para conversão	Não	Não	-	Sim
traduz a interface para uma tecnologia portátil	Sim	Sim	-	Sim
suporte para acesso a banco de dados	Não	Sim	Sim	Não
a interface traduzida pode ser migrada para diferentes linguagens de programação	Não	Não	-	Sim



Conclusões

- ▶ Auxílio na migração de aplicações Delphi
- ▶ Utiliza biblioteca multi-plataforma e multi-linguagem
- ▶ Possibilidade de migração para outras linguagens de programação



Extensões

extensão	complexidade
Integrar a ferramenta ao motor de script Pascal Script de forma a flexibilizar a construção de rotinas para tradução de componentes complexos, dispensando a necessidade de recompilar a aplicação.	10
Para componentes da VCL que não podem ser traduzidos segundo as especificações da Libglade, verificar a possibilidade de gerar código capaz de criar os elementos gráficos em tempo de execução.	10
Criar uma extensão para permitir gerar interfaces gráficas para o formato suportado pela classe GladeXML da GTK+.	8
Desenvolver uma interface gráfica para auxiliar na construção de arquivos de mapeamento de componentes, propriedades e eventos.	8
Criar esquemas XML para validação dos mapas de tradução para realizar a validação na aplicação, diminuindo o risco de erros.	8
Gerar código para carregamento da interface gráfica traduzida para diferentes linguagens de programação que suportam a GTK+.	7
Realizar a tradução do componente TFrame da biblioteca VCL.	1

