

Implementace GUI SCStudio

Bc. Milan Malota

Faculty of Informatics, Masaryk University

7. listopad 2012



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- vědět, co musí grafické rozhraní umět
- jsou jednotlivé položky dostatečně srozumitelné
- rozmyslet do jakého oddílu nové GUI začlenit

Znalosti na pozadí

- využíváme rodinu funkcí a definovaných elementů z ATL¹
 - nejsou svázány licenčními podmínkami
 - spousta užítku s minimem ručně psaného kódu
 - jednotlivé prvky jsou v hlavičkových souborech atl*.h
 - slabší dokumentace
- dialog třídy pojmenováváme s koncovkou Dlg
- k tvorbě využíváme rozhraní Visual Studio
- hodnoty se ukládají ve Windows Registry²

¹Active Template Library

²tohle nás již moc nezajímá vše je v registryManager.h

Model dialogu

- nejprve je nutné vymodelovat nový dialog
 - ve Visual Studiu, dále VS, otevřeme Resource View³
 - ve scstudio → dllmodule.rc nalezneme klíč Dialog
 - klikneme na něj pravým tlačítkem myši a zvolíme AddResource → New
 - otevře se nám prázdné okno
 - pokud bude dialog součástí hlavního Options pak je nutné nastavit
 - Style = Child
 - Border = None
 - TitleBar = False
 - a smazat potvrzovací tlačítka

³View → Resource View

Model dialogu II

- pokud máme vytvořený nový dialog můžeme začít vkládat jednotlivé elementy dialogu
 - elementy nalezneme v kartě Toolbox, jednoduše je lze přetáhnout na potřebné místo
 - každý z elementů má své Properties
 - jednotlivé elementy identifikujeme velkými písmeny podle toho, co vyjadřují⁴
- pokud je GUI vymodelované a prvky pojmenované, můžeme se pustit do implementace chování

⁴př.: `IDC_VALUE_INSTANCE_LENGTH`

Model dialogu III

Příklad: *IDD_BEAUTIFY_SETTING*

The image shows a configuration window for a dialog box. On the left, a tree diagram represents the dialog's structure, with nodes connected by lines and arrows. On the right, a settings panel is divided into sections:

- Instances length**:
 - const value: Sample ϵ mm
 - minimal
 - original value
 - align bottom
- Message position algorithm**:
 - linear programming solver
 - heuristic methods
- Other settings**:
 - Length of lost/found message: Sample ϵ mm
 - Slope of messages: Sample ϵ mm

Implementace chování - hlavičkový soubor

- nyní vytvoříme třídu simulující chování dialogu
 - pro začátek můžeme zkopírovat některou z již existujících tříd
 - třída T musí dědit z `ATL::CSettingsPageDialog` a `CWinDataExchange<T>`
 - public members
 - enum IDD od dialogu, ke kterému se třída vztahuje
 - konstruktor bere `Visio::IVApplicationPtr vsoApp` a naplňuje hodnoty rodiče⁵
 - destruktor

⁵`TDlg(Visio :: IVApplicationPtr vsoApp) :
CSettingsPageDialog(IDD, true, TDlg :: StartDialogProc), m_vsoApp(vsoApp)`

Implementace chování - hlavičkový soubor II

- public members II
 - dále určíme cestu v registrech, kde se nám budou ukládat hodnoty
 - vytvoříme si proměnné ve kterých se budou jednotlivé hodnoty ukládat během práce s dialogem
 - editbox → float *m_my_name_value*
 - checkbox → int *m_my_name*
 - radiobutton → long *m_use_my_name*

Příklad: mějme editbox do něhož uživatel ukládá hodnotu délky ztracené/nalezené zprávy. Podle společné konvence jsme jej nazvali kupříkladu: *IDC_VALUE_INCOMPLETE_MESSAGE*, hodnota s kterou operujeme je typu float, a proto si definujeme: float *m_length_incomplete_msg_value*.

Implementace chování - hlavičkový soubor III

- private members
 - v privátních atributech jsou definovány handlers
 - a objekty dialogu⁶
 - v *BEGIN_DDX_MAP(TDlg)* nalezneme různé HANDLERY v závislosti na objektu s kterým pracujeme
 - *DDX_CONTROL_HANDLE* → přiřazuje název radio buttonu k objektu CButton
 - *DDX_FLOAT_RANGE* → handler, který automaticky testuje, zda hodnota v edit boxu splňuje daná omezení
 - *DDX_FLOAT* → přiřazuje název editboxu k proměnné typu float
 - a spousta dalších...
 - mapu ukončíme *END_DDX_MAP*

Příklad: již jsme řekli, že máme definované *IDC_VALUE_INCOMPLETE_MESSAGE* a *m.length_incomplete_msg_value*, a chceme hodnotu v editboxu svázat s touto proměnnou.

⁶př.: CButton, CEdit, CUpDownCtrl

Implementace chování - hlavičkový soubor III

- private members
 - v privátních atributech jsou definovány handlers
 - a objekty dialogu⁶
 - v *BEGIN_DDX_MAP(TDlg)* nalezneme různé HANDLERY v závislosti na objektu s kterým pracujeme
 - *DDX_CONTROL_HANDLE* → přiřazuje název radio buttonu k objektu CButton
 - *DDX_FLOAT_RANGE* → handler, který automaticky testuje, zda hodnota v edit boxu splňuje daná omezení
 - *DDX_FLOAT* → přiřazuje název editboxu k proměnné typu float
 - a spousta dalších...
 - mapu ukončíme *END_DDX_MAP*

Příklad: již jsme řekli, že máme definované *IDC_VALUE_INCOMPLETE_MESSAGE* a *m.length.incomplete_msg_value*, a chceme hodnotu v editboxu svázat s touto proměnnou. Řešení:

- pomocí *DDX_FLOAT_RANGE* lze například otestovat, zda je číslo kladné
- pomocí *DDX_FLOAT* přiřadíme k *IDC_VALUE_INCOMPLETE_MESSAGE* hodnotu typu float

⁶př.: CButton, CEdit, CUpDownCtrl

Implementace chování - hlavičkový soubor III

- private members II
 - mapa zpráv, neboli funkcí, které se provedou při nějaké akci s dialogem
 - typicky otevření okna, co se má udělat
 - pokud někdo klikne na nějaké radio, co má udělat
 - v *BEGIN_MSG_MAP(TDlg)* nalezneme různé HANDLERY v závislosti na objektu s kterým pracujeme
 - *MESSAGE_HANDLER* → typicky pro vytvoření funkce OnInitDialog
 - *COMMAND_ID_HANDLER* → přiřazuje k názvu nějakého radia funkci, která se má vykonat, pokud systém detekuje, že se s radiem něco stalo, př.: byl označen
 - a spousta dalších...
 - mapu ukončíme *END_MSG_MAP*

Implementace chování - hlavičkový soubor IV

- private (definice funkcí dialogu)
 - dále definice funkcí:
 - `voidOnDataValidateError(UINTnCtrlID, BOOLbSave, _XData&data);`
 - `voidOnDataExchangeError(UINTnCtrlID, BOOLbSave);`
 - definice `LRESULTOnInitDialog(UINTuMsg, WPARAMwParam, LPARAMlParam, BOOL&bHandled);`
 - pokud je dialog členem hlavního nastavení pak musíme přepsat `OnApply` a `OnHelp()`
 - dále definice všech funkcí které jsme nedefinovali v `MSG_MAP`

Implementace metod

- implementace definic z hlavičkového souboru
- *OnInitDialog*
 - co se děje pokud se má vytvořit naše GUI
 - nejprve dokončíme nalepení jednotlivých prvků dialogu na objekty definované v hlavičce, které jsme nenalepili pomocí *DDX_MAP(TDlg)*
 - Příklad: přiřazení spinboxu, využíváme Attach; vytvoření tooltipu

Implementace metod

- implementace definic z hlavičkového souboru
- *OnInitDialog*
 - co se děje pokud se má vytvořit naše GUI
 - nejprve dokončíme nalepení jednotlivých prvků dialogu na objekty definované v hlavičce, které jsme nenalepili pomocí *DDX_MAP(TDlg)*
 - Příklad: přiřazení spinboxu, využíváme *Attach*; vytvoření tooltipu
 - načteme si hodnoty z registrů (*LoadRegistryData()*);
 - uložíme hodnoty do dialogu pomocí *DoDataExchange(FALSE)*;
 - a zavoláme *UpdateControls()*;

Děkuji za pozornost



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ