

INTERACTIVE DESIGN OF USER INTERFACE

Libor LAICHMANN, Master Degree Programme (5)
Dept. of Intelligent Systems, FIT, BUT
E-mail: xlaich00stud.fit.vutbr.cz

Supervised by: Ing. Vladimír Marek

ABSTRACT

This work deals with the methods of user interface design. It presents top-down and bottom-up methods to design UI. The use of these two methods is also studied. A desktop application supporting those methods is currently in implementation phase. The current state, some screenshots, and future visions of the application are also reported.

1 ÚVOD

Informační systémy se většinou implementují jako třívrstvé aplikace. Tyto vrstvy jsou:

- Vrstva uživatelského rozhraní (dále jen UI)
- Aplikační vrstva
- Datová vrstva

Pro návrh aplikační a datové vrstvy existují známé vizuální a modelovací nástroje či jazyky, např. UML (Unified Modeling Language), ERD (Entity Relationship Diagram), DFD (Data Flow Diagram) a další. Tyto nástroje mají za úkol mimo jiné zefektivnit vývoj těchto vrstev. Naše práce si bere za cíl vytvořit model a na něm založenou aplikaci umožňující právě takové zefektivnění pro vrstvu UI. Jejím účelem je nejen možnost snadného a transparentního vytváření UI ale také generování scénářů užití, popřípadě celých prototypů částí informačního systému. Její další nespornou výhodou je fakt že výsledná data UI (XML) jsou nezávislá na výsledném programovacím jazyku ve kterém je IS realizován.

Ačkoliv implementace uživatelského rozhraní nepatří mezi nejobtížnější programátorské disciplíny, jedná se o práci, která zabírá velké procento realizace celého informačního systému a její zefektivnění pomocí vhodného nástroje může výrazným způsobem zkrátit dobu realizace a ušetřit náklady. Navíc možnost vytváření scénářů a prototypů umožní zadavateli získat informace o vzhledu a chování požadovaného IS.

V této zprávě dále naleznete:

- Navržené metodiky pro návrh formulářových oken: shora dolů a zespona nahoru
- Informace o implementaci nástroje pro podporu vizuálního modelování UI

2 METODY NÁVRHU FORMULÁŘOVÝCH OKEN IS

Konzistence vzhledu formulářových oken (dále jen formulářů) IS přináší řadu výhod jak pro uživatele (intuitivnost, snadné pochopení), tak i pro implementaci (znovupoužitelnost komponent, snadná údržba). Konzistence spočívá například v tom, že prvky stejného významu opakující se na více formulářích, jsou umístěny ve stejných pozicích (např. aplikační menu). Obdobně prvky, které mají shodný způsob zadání vstupní hodnoty (např. výběrem z jiného formuláře), budou vypadat obdobně.

Konzistence návrhu je zajištěna sdílenými definicemi společných rysů, na něž se jednotlivé formuláře a jejich prvky odkazují. Změny společné definice se projeví ve všech prvcích, které se na ni odkazují.

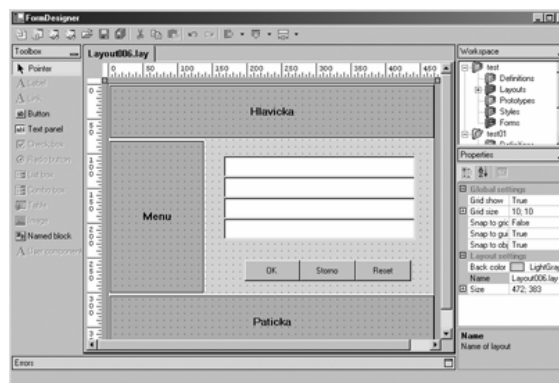
Byly navrženy 2 základní metodiky pro návrh formulářů, které je možno libovolně kombinovat:

- Návrh shora dolů (top-down)
- Návrh zespodu nahoru (bottom-up)

2.1 NÁVRH SHORA DOLŮ

Tato metodika vychází z rámcového rozložení jednotlivých oblastí na formuláři, které postupně zpřesňuje o jejich obsah. Do návrhu formuláře (tzv. *layoutu*) je možné kromě základních formulářových prvků (jednořádkový text, bombo box, tlačítko, apod.) vložit tzv. *pojmenované bloky*. Pojmenované bloky jsou oblasti, jejichž obsah není na této úrovni návrhu formuláře definován. Obsah pojmenovaného bloku je specifikován na úrovni tzv. *define*, která mapuje do tohoto bloku jiný návrh formuláře. K jednomu layoutu je možno přidat více definic, a tak získat z jednoho rámcového rozložení celou skupinu vzhledově obdobných formulářů.

Definice lze odvozovat z jiných definic, podobným způsobem, jakým funguje dědičnost v objektově orientovaných jazycích. Odvozená definice přebírá kompletní nastavení rodičovské definice, přičemž může své vlastnosti předefinovat.

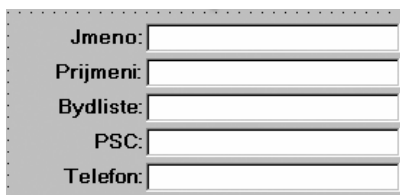


Obr. 1: Příklad navržení formuláře s pojmenovanými bloky v navrhované aplikaci

Obdobný princip návrhu shora dolů je podporován frameworkem Tiles (součást Apache Struts), který pracuje na platformě J2EE.

2.2 NÁVRH ZESPODU NAHORU

Tato metodika umožňuje definovat opakovaně používané komponenty složené z více formulářových prvků (případně dalších komponent). Jako příklad může sloužit zadání adresy:



The image shows a vertical stack of five text input fields. Each field is preceded by a label: 'Jmeno:', 'Prijmeni:', 'Bydliste:', 'PSC:', and 'Telefon:'. The fields are contained within a light gray rectangular frame with a dotted border.

Obr. 2: *Příklad komponenty pro metodiku zespona nahoru*

Tuto vizuální komponentu lze vkládat do jednotlivých návrhů formulářů. Eventuelní změna komponenty Adresa se automaticky projeví ve všech jejích výskytech.

3 IMPLEMENTACE

Nástroj, který je v současné době v implementační fázi, podporuje obě výše uvedené metodiky. Je realizován jako desktopová aplikace běžící na platformě .NET. Mimo uvedené metodiky aplikace podporuje či plánuje další rysy:

- *Styly* – lze definovat jednotlivé styly, které lze přiřazovat definicím. Změna stylu pak globálně změní vizuální vlastnosti všech zobrazovaných prvků.
- *Validate* – k jednotlivým polím lze připojovat informace o tom, jaký rozsah hodnot do nich může uživatel vkládat (povinné pole, numerické, ověření na regulární výraz, apod.)
- *Scénáře* – aplikace také umožňuje definici jednotlivých scénářů, případu užití. Tedy po které uživatelské akci následuje který formulář, apod.
- *Prototypy* – ze scénářů je možné generovat jednoduché demonstrace, prototypy uživatelského rozhraní, dle konkrétního scénáře, případně globální prototyp uživatelského rozhraní celého informačního systému.

4 SOUČASNÝ STAV A DALŠÍ VIZE

V současné době obsahuje nástroj podporu pro návrh shora-dolů a zespona nahoru. Pokračují práce na definici stylů a validací. Scénáře a prototypy zatím nejsou implementovány.

Po dohotovení podpory scénářů a prototypů tento nástroj povýší na výkonný analytický nástroj, jehož výstup bude sloužit také jako podklad pro implementaci informačního systému.

LITERATURA

- [1] Laichmann, L.: Editor Formulářů pro IS, ročníkový projekt, Brno, 2005
- [2] Gamma, E., aj.: Návrh programů pomocí vzorů, Grada, 2003, ISBN 80-247-0302-5
- [3] Schmuller, J.: Myslíme v jazyku UML, Grada, 2001, ISBN 80-247-0029-8