Praktické aspekty vývoje software

Existující komponenty, knihovny, make, sestavení programů a závislosti, SWIG

Ing. Jan Kouřil, Ing. Jaroslav Dytrych

Fakulta informačních technologií

22. února 2016

Obsah

Identifikace dostupných komponent

Co je třeba zohlednit:

- Licence – dostupný zdrojový kód, možnosti přizpůsobení, cena, licence výsledného produktu
- Výkon
- Dokumentace
- Přenositelnost (operační systémy, 32 vs 64 bitů, ARM, ...)
- Znalost knihovny vs. požadovaná funkcionalita
- Volba programovacího jazyka (vysokoúrovňové programovací jazyky)

Existující kompilátory

- Unix - GCC, CLANG
- Windows - MinGW, MSVC

Dostupné komponenty

Hotové produkty

- Přidání požadovaných funkcí do existujících produktů (moduly, rozšíření)
- Přístup přes API (databáze, webové služby)

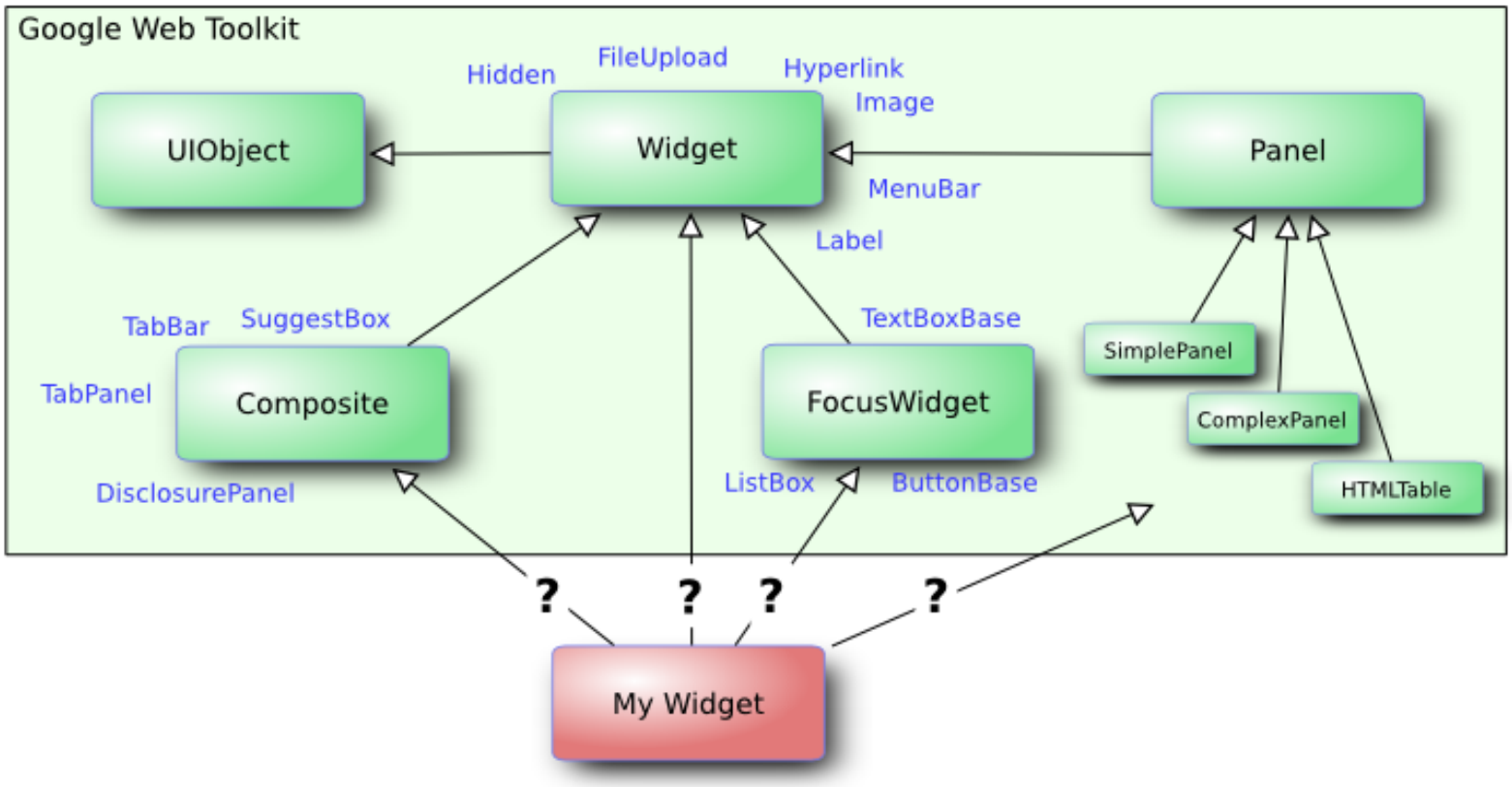
Frameworky

- Určuje chování aplikace, které programátor pouze upravuje
- Většinou GUI, webové aplikace

Knihovny

- Chování aplikace určuje programátor a volá funkce z knihovny
- Nelze vždy určit, zda se jedná o knihovnu, či framework

Příklad frameworku (Google Web Toolkit)



Licence

Vybrané open source licence

Licence	Linkování s uzavřenou aplikací	Distribuce "výtvoru"	Distribuce upraveného kódu
GPL	Ne	GPL komp.	GPL
LGPL	Ano	s omezením	GPL, LGPL
Apache	Ano	Ano	Ano
BSD	Ano	Ano	Ano

GTK+

Vlastnosti knihovny

- GUI
- Napsáno v C
- Možno použít i v C++, Pythonu, ...
- Linux, Unix, Mac OS X, MS Windows
- GNU LGPL 2.1

Příklady programů

- Gimp
- Google Chrome
- Pidgin
- Firefox

Qt

Vlastnosti frameworku

- GUI, sítě, XML, databáze ...
- Napsáno v C++
- Možno použít i v Pythonu, Ruby ...
- Linux, Mac OS X, MS Windows, Sailfish, Android, iOS, Blackberry
- Plán - Windows Phone, WinRT
- Komerční licence, GNU LGPL 2.1, GPL 3.0

Moduly

- QtCore, QtGui, QtWidgets
- QtMultimedia, QtNetwork, Qt QML, Qt Quick
- Qt SQL, Qt Test, Qt Webkit

Další knihovny

OpenCL

- Paralelní programování

OpenCV

- Zpracování obrazu, počítačové vidění, načítaní různých obrazových formátů ...

OpenGL

- Grafická knihovna pro programování ve 3D

curl a libcurl

- Přenos dat přes HTTP, HTTPS, FTP, SFTP, IMAP ...

Statické × dynamické linkování

Statické linkování

- Použité části knihovny se při linkování stávají součástí programu
- K provozu stačí přeložený program

Dynamické linkování

- Knihovna je přeložena zvlášť
- Při linkování se ukládají pouze odkazy na symboly definované v dynamické knihovně
- K provozu je třeba přeložený program a knihovna
- Šetří paměť a místo na disku
- Knihovna je v paměti jen jednou a může ji využívat více programů
- Problémy s verzemi knihoven a programů
- `*.dll` ve Windows, `*.so` v Linuxu, `*.dylib` na MacOS

GNU Make

make

- Automatický překlad (nejen) programů
- Nezávislý na jazyku
- Multiplatformní
- Umožňuje překlad a instalaci uživatelům bez znalosti struktury projektu
- Úkolem programu make (a jeho pokračovatelů, viz např. scons s pythonovskou syntaxí) je poznat, které části velkého projektu mají být při určité změně znovu generovány (obvykle přeloženy), a spustit příslušné příkazy, které generování provedou
- Makefile je souborem definic cílů (targets), předpokladů (prerequisites) a příkazů pro zpracování jednotlivých cílů
- velmi dobře dokumentuje, jak který soubor vznikl ...

GNU Make

Soubory

- `makefile`
- `Makefile`

Použití

```
$ make [cíl]
$ make -B [cíl]
$ make -n [cíl]
$ make -p [cíl]
```

- -B vykoná vše (znovu přeloží i to, co je již přeložené)
- -n nevykonává, jen vypisuje co by se dělo
- -p k tomu navíc vypíše nastavení proměnných, implicitních pravidel, ...

Pravidla

Makefile

```
hello: hello.c
    gcc hello.c -o hello
```

Obecně

```
cíl_1 cíl_2 cíl_n : závislost_1 závislost_2 závislost_n
    příkaz_1
    příkaz_n
```

Jak to funguje

- Najde první cíl
- Aktualizuje závislosti (najde odpovídající cíle)
- Provede příkazy (musí být odsazeny tabulátorem)
- Pro každý řádek příkazů se spustí nový shell (interpret příkazů), pokud chceme, aby se příkazy ovlivňovaly, napíšeme je za sebou

Proměnné

Makefile

```
CXXOPT = -O3
CXXFLAGS = $(CXXOPT) -ansi -Wall -lm \
    `pkg-config --cflags --libs opencv`
```

Automatické proměnné

- $@ cíl
- $< první závislost
- $? všechny závislosti novější než cíl
- $^ všechny závislosti bez duplicit
- $+ všechny závislosti včetně duplicit
- $(@D) jen adresář
- $(@F) jen jméno souboru

vpath

Makefile

```
vpath %.cpp src
vpath %.h  src
vpath %.o  obj

%.o : %.cpp
    $(CPP) -c $< -o obj/$@ $(CXXFLAGS)
```

Pravidla pro více cílů

Makefile

```
$(BIN): main.o EA.o RBN.o Picture.o IniFile.o
$(BIN1): interpret.o  RBN.o Picture.o IniFile.o
$(BIN2): addnoise.o  Picture.o IniFile.o
$(BIN) $(BIN1) $(BIN2) :
    cd obj; $(CPP) $(@F)  -o ../$@ $(CXXFLAGS)
```

.PHONY

- jsou provedeny vždy, nekontroluje se existence souboru

Makefile

```
.PHONY: pack clean debug

debug:
    make -B all "CXXOPT=-g3 -DDEBUG"

clean:
    rm ./obj/*.* $(BIN) $(BIN1) $(BIN2)

pack:
    tar -cvzf $(PROJECT_NAME).tar.gz $(FILES)
    zip  $(PROJECT_NAME).zip $(FILES)
```

CMake

cmake

- Generuje Makefile
- Konfigurační soubor CMakeFileLists.txt

Možnosti CMake

- Základní nastavení
- Přidání knihoven
- Instalace a testy
- Zjištění nainstalovaných funkcí
- Přidání generovaných souborů
- Tvorba instalátoru

Skript ./configure

Potřebné soubory

- autogen.sh
- configure.ac
- Makefile.am
- NEWS README AUTHORS ChangeLog

Sestavení skriptu, přeložení, instalace

- 'autoreconf –force –install' - spustí postupně aclocal, autoconf, autoheader, automake
- Vygenerují se config.h.in a Makefile.in
- './configure' - vytvoří Makefile
- 'make' - přeloží program
- 'make install' - nainstaluje program do zvolené cesty

GTK+

GTK+

- widgety
- kontejnery
- `gtk_main()`
- signály
- handlery

Hierarchie objektů

```
GObject
+----GInitiallyUnowned
     +----GtkObject
          +----GtkWidget
               +----GtkContainer
                    +----GtkBin
                         +----GtkWindow
```

Glade

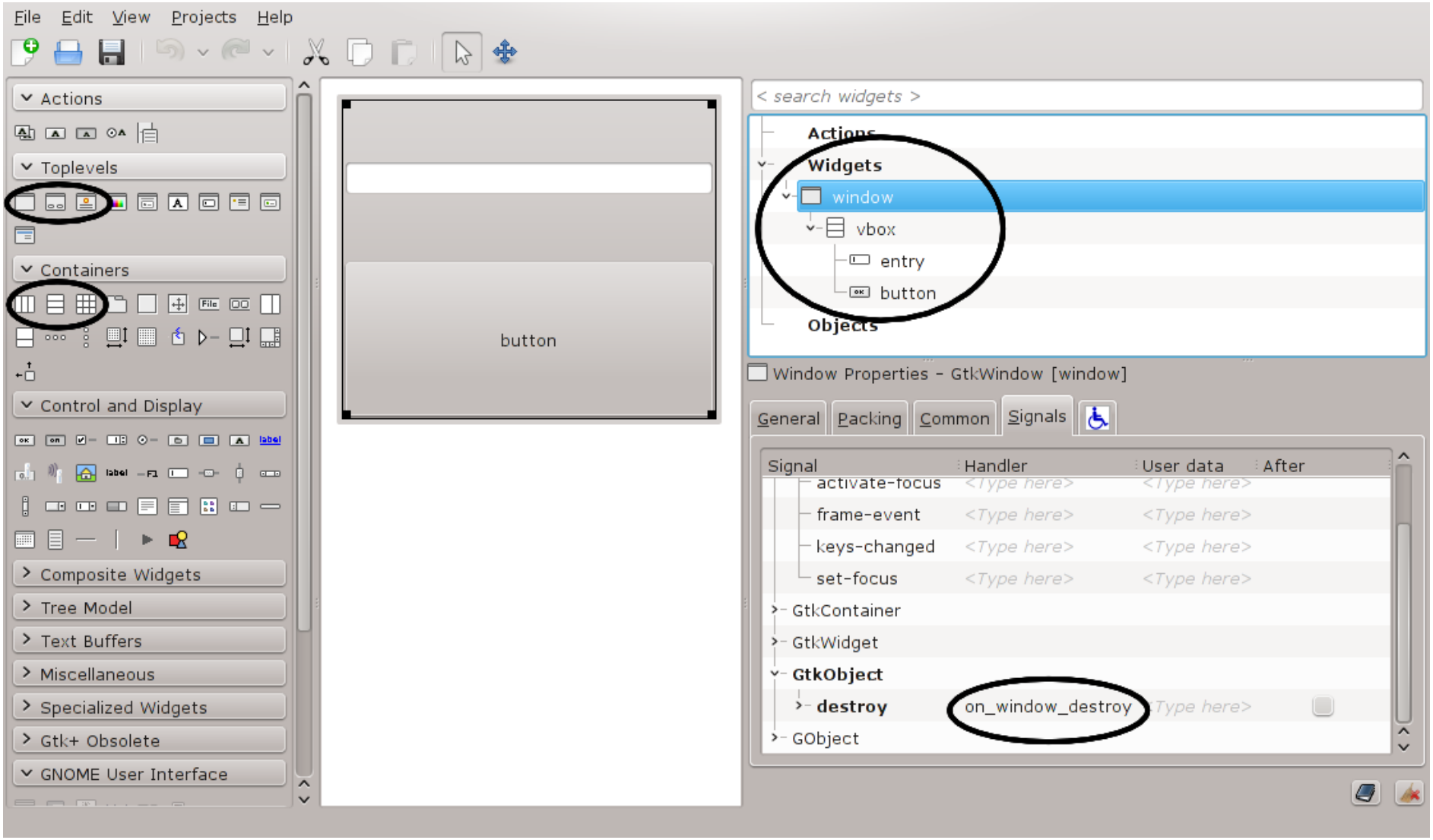
Návrh GUI

- Grafický
- Oddělený od aplikace
- Různé GUI pro jednu aplikaci bez nutnosti opětovného překladu
- Je popsáno XML souborem

GtkBuilder

- Vytvoří GUI z XML souboru

Návrh GUI pomocí Glade



Překlad pomocí pkg-config

pkg-config

- automaticky nastaví volby překladače

Použití

- zjištění verze GTK

```
pkg-config --modversion gtk+-3.0
```

- volby potřebné pro překlad

```
pkg-config --cflags gtk+-3.0
```

- volby potřebné pro linkování

```
pkg-config --libs gtk+-3.0
```

- spuštění překladače

```
gcc -Wall -g -o tutorial main.c \
-export-dynamic `pkg-config --cflags --libs gtk+-3.0`
```

SWIG

SWIG (Simplified Wrapper and Interface Generator)

- Propojení C a C++ se vysokoúrovňovými jazyky
- Podpora pro C#, D, Go, Java, Lua, Octave, Perl, PHP, Python, Ruby
  ...

SWIG Příklad

example.c

```
#include <time.h>
double My_variable = 3.0;
int fact(int n) {
    if (n <= 1) return 1;
    else return n*fact(n-1);
}
int my_mod(int x, int y) {
    return (x%y);
}
char *get_time()
{
    time_t ltime;
    time(&ltime);
    return ctime(&ltime);
}
```

SWIG Příklad

example.i

```
%module example
 %{
 /* Put header files here or function declarations*/
 /* like below */
 extern double My_variable;
 extern int fact(int n);
 extern int my_mod(int x, int y);
 extern char *get_time();
 %}

 extern double My_variable;
 extern int fact(int n);
 extern int my_mod(int x, int y);
 extern char *get_time();
```

SWIG Python

Překlad

```
$ swig -python example.i
$ gcc -c example.c example_wrap.c \
        -I/usr/include/python2.7 -fPIC
$ ld -shared example.o example_wrap.o -o _example.so
```

Použití

```
>>> import example
>>> example.fact(5)
120
>>> example.my_mod(7,3)
1
>>> example.get_time()
'Sun Feb 11 23:01:07 1996'
>>>
```

Literatura I


http://en.wikipedia.org/wiki/Not_Invented_Here.


http://en.wikipedia.org/wiki/Software_framework.


http://developer.kde.org/.../licenses_summary.html.


Robert Mecklenburg.
Managing Projects with GNU Make (Nutshell Handbooks).
O'Reilly Media, 3 edition, November 2004.


http://www.gnu.org/software/make/.


http://www.fit.vutbr.cz/ martinek/clang/make.html.


http://www.micahcarrick.com/gtk-glade-tutorial-part-1.html.


http://library.gnome.org/devel/gtk/.


http://www.swig.org/tutorial.html.