XPCE 6.0.0 Grafikus felületek Prologhoz

Tóth Balázs György
tbalazs@sch.bme.hu

VFLP 2003.

Tartalom

- Grafikus felületek
- Illesztés a prologhoz
- XPCE – Első találkozás
- Dialógusok
- Egyszerű grafika
- Érdekességek
- Összefoglalás

Grafikus felületek általában

- Segítik a programok használatát
- Intuitívabb felület, mint a parancssor
- Jobban áttekinthető

Grafikus felületek Prologhoz

- Natív grafikus könyvtárak
  - Tetszőleges grafikus API
  - C/C++ felület
  - A valós logika Prologban
  - Nehézkes adatcsere
    - Nem természetes Prolog típusok
  - Nagy overhead

Grafikus felületek Prologhoz

- Külső GUI nyelvek (Tcl/TK)
  - Direkt GUI készítésére lettek kitalálva
  - Pipe-ok, beágyazás
  - Természetesebb, mint a natív API
  - Még mindig nem elég hatékony!

Grafikus felületek Prologhoz

- Prologban megvalósított GUI (XPCE)
  - Természetes adatreprezentáció
  - Kicsi overhead
  - Ahol fut a prolog ott fut a GUI is
  - Objektum-orientált mag
  - Prolog könyvtár!

Az XPCE rövid története

- „X-Window version of the Portable Computing Environment”
- 1985-ben a PCE első verzióját termodinamikai rendszerek vizualizációjához fejlesztették ki
- A mag nagy részét újraírták Quintus- később SWI Prologhoz
- A 4.0-ás verzió után az addigi SunView helyett X-Window rendszert használták
- A 4.7-es verzió óta Win32 kompatibilis
- Az 5.0-ás verzió javított kapcsolódási felülettel jelent meg, mely lehetővé tette a natív adatcserét a Prolog-gal.
- Az 5.2-es verzió után GPL-2 licenszelésű lett ac XPCE
- A 6.0-ás verzió legnagyobb előnye az LGPL licensz, mely lehetővé teszi a kereskedelmi célú felhasználást!

Illesztés a prologhoz

- A megvalósítás prolog könyvtárként áll rendelkezésre
- A létrehozott objektumok referenciákon keresztül érhetők el
- A prolog adatszerkezetek konvertálhatók XPCE adattípusokká

Illesztés a prologhoz

- Az objektumok létrehozása egy összetett kifejezés segítségével történik
  - Functor(...InitArg...)
    - A Functor határozza meg a létrehozandó osztályt
    - Az InitArg határozza meg a kezdeti paramétereket
      - XPCE adat objektummá konvertálódik a megfelelő szabályok segítségével
      - Közvetlen prolog predikátumokat is használhatunk!

Illesztés a prologhoz

- A különböző prolog megvalósításokkal való kompatibilitás miatt két új predikátum
  - **require**(*:ListOfNameArity*)
    - A szükséges predikátumok listája
  - **auto_call**(*:Goal*)
    - Automatikusan betölti a szükséges predikátumokat, ha a prolog megvalósítás támogatja az automatikus betöltést

Az első lépések

- Az XPCE betöltése
  - :- use_module(library(pce)).
- Objektumok
  - Objektum-orientált grafikus mag
  - Az alap építőkövek az objektumok
  - Négy alapvető predikátum az objektumok manipulálására

Az első lépések

- new/2
  - new(?Reference, +NewTerm).
  - Objektum létrehozása
  - Példa:
    - ?- new(@demo, dialog('Demo Window')).
    - ?- new(P, point(10,20)).

      P = @772024

Az első lépések

- send/2
  - send(+Receiver, +Selector(...Args...)).
  - Az objektum állapotának megváltoztatása.
  - Példa
    - ?- send(@772024, x(15)).
    - ?- send(@demo, append(text_item(name))).
    - ?- send(@demo, open).

Az első lépések

- get/2
  - get(+Receiver, +Selector(...Args...), ?Return).
  - Az objektum állapotának lekérdezése
  - Példa
    - ?- get(@772024, y, Y).

      Y = 20
    - get(@demo, display, D).

      D = @display/display

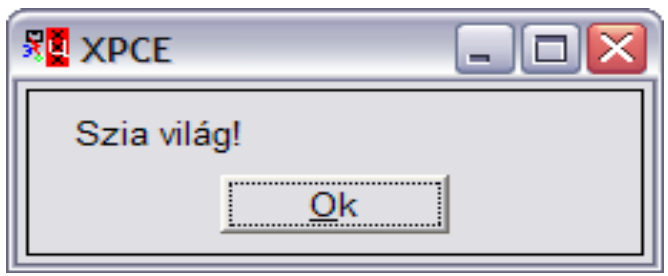
Az első lépések

- free/1
  - free(+Receiver).
  - Az objektum felszabadítása
  - Példa
    - ?- free(@772024).
    - ?- free(@demo).
    - ?- free(@display).

Az első lépések

- Szia világ!

```
szia_vilag :-
        new(D, dialog('XPCE')),
        send(D, append, label('macko','Szia világ!')),
        send(D, append, new(B, button(ok, message(D, destroy)))),
        send(D, open).
```



Az első lépések

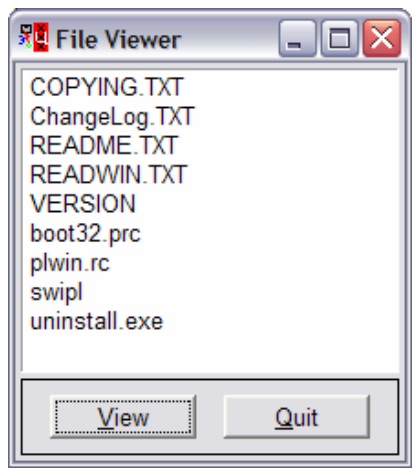
- Egy összetettebb példa

```
:- use_module(library(pce)).
:- use_module(library(help_message)).

% File viewer example

file_viewer(Dir) :-
        new(F, frame('File Viewer')),
        send(F, append(new(B, browser))),
        send(new(D, dialog), below(B)),
        send(D, append(button(view, message(@prolog, view, B?selection?key)))),
        send(D, append(button(quit, message(F, destroy)))),
        send(B, members(directory(Dir)?files)),
        send(F, open).

view(F) :-
    send(new(V, view(F)), open),
    send(V, load(F)).
```



Az első lépések

- Egy összetettebb példa
  - send(@prolog, view, B?selection?key)
    - @prolog: a gazda prolognak adja a vezérlést
    - ?(?(B, selection), key): a felületelem elérésére
  - A felület elemeket automatikusan is el lehet helyezni, de lehetőség van explicit módon is megmondani a helyüket.

Szintaktikus édesítőszerek

- Opcionális argumentumok
  - @default, az alapértelmezett érték
  - ?- new(X, style(@default, @default, @default, @default, @on)).
  - ?- new(X, style(underline := @on)).
  - area->set: x=[int], y=[int], width=[int], height=[int]
  - ?- new(A, area),
    send(A, set(y := 10, height := 50)).

Szintaktikus édesítőszerek

- Automatikus típuskonverzió
  - Sok esetben az objektum automatikusan létrejön
  - ...,
    send(Box, colour(colour(red))),
    ...
  - ...,
    send(Box, colour(red)),
    ...

  A két példa ugyanazt a hatást váltja ki!

Szintaktikus édesítőszerek

- send/[2-12] és get/[3-13]
  - Egyszerre több paramétert is át lehet adni
  - Kompatibilitás az előző verziókkal
  - Néhol kényelmesebb használni
  - Példa
    - send(Box, width(100)).
      send(Box, width, 100).
    - get(Point, distance(point(10,10), D)).
      get(Point, distance, point(10,1), D).

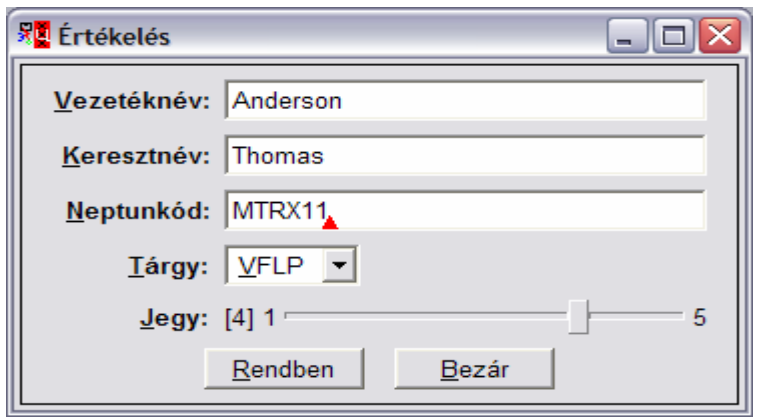
Dialógusok

- A dialógusok összeállításához beépített elemek állnak a rendelkezésünkre
  - button: egyszerű nyomógomb
  - text_item: szövegbeviteli mező
  - int_item: számbeviteli mező
  - slider: értékkiválasztó csúszka
  - menu: menü
  - menu_bar: legördülő menü
  - label: felirat
  - list_browser: egy lista elemeit jeleníti meg
  - editor: szövegszerkesztő
  - tab, tab_stack, dialog_group: több fülből álló dialógusok támogatása

Dialógusok példa

```
ertekeles :-
        new(D, dialog('Értékelés')),
        send(D, append, new(VN, text_item(vezetéknév))),
        send(D, append, new(KN, text_item(keresztnév))),
        send(D, append, new(Neptun, text_item(neptunkód))),
        send(D, append, new(Targy, menu(tárgy, cycle))),
        send_list(Targy, append, ['Dp', 'NHLP', 'VFLP']),
        send(D, append, new(Jegy, slider('Jegy', 1, 5, 5))),
        send(D, append, button(rendben, and(message(@prolog, tarol,
                                        VN?selection,
                                        KN?selection,
                                        Neptun?selection,
                                        Targy?selection,
                                        Jegy?selection),
                                message(VN, clear),
                                message(KN, clear),
                                message(Neptun,clear),
                                message(Targy, selected, 'Dp', true),
                                message(Jegy, selection, 5)))),
        send(D, append, button(bezár, message(D, destroy))),
        send(D, open).

tarol(Vezeteknev, Keresztnev, Neptunkod, Targy, Jegy) :-
        format('Tárolva: ~w ~w (~w), ~w: ~w~n',
             [Vezeteknev, Keresztnev, Neptunkod, Targy, Jegy]).
```



Felületelemek elhelyezése

- Automatikus elhelyezés, gyakran nem kielégítő
- Manuális elhelyezés
  - Pixelpontosan
    - Problémák lehetnek átméretezésnél
    - Nem javasolt!
  - Logikailag
    - Egyszerűbb a tervezés
    - Alkalmazkodik a megjelenítéshez

Felületelemek elhelyezése

- Logikai definíció
  - Egy logikai rács segítségével történik
  - Például
    - dialog_item->above
    - dialog_item->below
    - dialog_item->left
    - dialog_item->right
  - Sok lehetőséget nyújt, és könnyebb átlátni

Egyszerű grafika

- Eddig felületelemeket használtunk
- Mi magunk is rajzolhatunk!
  - Számítások megjelenítéséhez
  - Magyarázó rajzokhoz
  - Kicsit bonyolultabb, de sokkal rugalmasabb!

Egyszerű grafika

- „Picture” objektum
  - Ablak görgetősávokkal
  - Negatív és pozitív irányban is végtelen
  - Példa
    - ?- new(@p, picture('Demo Picture')),
      send(@p, open).

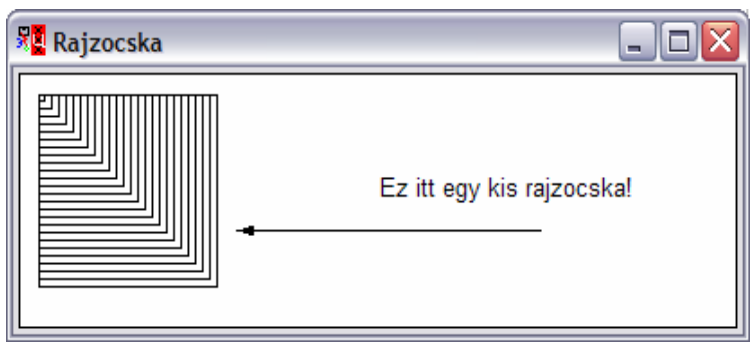
Egyszerű grafika

- Alapvető építőelemek
  - arrow: nyíl
  - bezier: Bezier görbe
  - bitmap: fekete-fehér és színes képek megjelnítésére
  - box: négyzet (lekerekített is lehet)
  - circle: kör
  - ellipse: ellipszis (kitöltött is lehet)
  - arc: ellipszis cikk (tortaszelet alakú is lehet)
  - line: egyenes (nyilak is lehetnek rajta)
  - path: több egyenesből álló út
  - text: szöveg (többféle stílusban)

Egyszerű grafika

```
rajzocska :-
        new(P, picture('Rajzocska')),
        dobozok(P),
        send(P, display, new(T, text('Ez itt egy kis rajzocska!')), point(200,50)),
        send(P, display, new(L, line(120, 80, 290, 80, first))),
        send(P, open).

dobozok(P) :-
        forall(between(0, 25, H),
        send(P, display, new(B, box(H*4,H*4)), point(10,10))).
```



Egyszerű grafika

- Összetett rajzok
  - Több grafikus elem összefogható
  - Egy üzenettel elérhetjük az összeset
  - Összetett osztályok
    - device: legáltalánosabb osztály, minden más ebből származik
    - figure: a device részhalmaza (vágás, háttér, stb)
    - format: elrendezés
    - table: a format egy változata, mely HTML kompatibils

Egyszerű grafika

- Kapcsolatok
  - Két grafikus objektumot összekapcsolhatunk logikailag
  - Bármely áthelyezésével a köztük lévő nyilak is változnak!
  - Kapcsolat osztályok
    - connection
    - handle
    - link
    - connect_gesture

Egyszerű grafika

- Grafikus kényszerek
  - A logikai elrendezés állandó marad
  - A kényszerekkel összekötött objektumok követni fogják egymást!

Egyszerű grafika

- Grafikai elemek aktiválása egérrel
  - Lehetőség van a grafikai elemekhez eseményeket kötni, melyeket egérrel aktiválhatunk (pl. rákattintunk)
  - Osztályok:
    - handler
    - handler_group
    - key_binding
    - click_gesture
    - connect_gesture
    - move_gesture
    - move_outline_gesture
    - resize_gesture
    - resize_outline_gesture

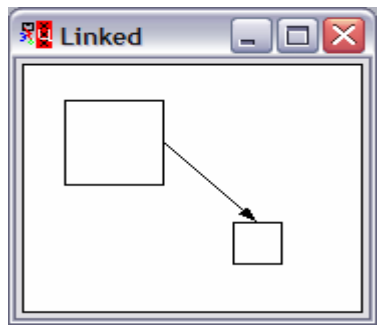
Egyszerű grafika

```
:- use_module(library(pce)).
:- use_module(library(help_message)).

:- pce_global(@in_out_link, make_in_out_link).

make_in_out_link(L) :-
                new(L, link(in, out, line(arrows := second))).

linked :-
      new(P, picture('Linked')),
      send(P, open),
      send(P, display, new(B1, box(50,50)), point(20,20)),
      send(P, display, new(B2, box(25,25)), point(100,100)),
      send(B1, handle, handle(w, h/2, in)),
      send(B2, handle, handle(w/2, 0, out)),
      send_list([B1, B2], recogniser, new(move_gesture)),
      send(B1, connect, B2, @in_out_link).
```



Érdekességek

- Az XPCE képes HTML kimenetet is előállítani
- A httpd könyvtár segítségével létrehozhatunk egy egyszerű webszervert

Összefoglalás

- A grafikus felületek segítségével nő a kifejező erő!
- Az XPCE természetes kiterjesztés a prologhoz!
- Nagy rendelkezésre álló eszköztár!
- Tetszőlegesen bővíthető!

Kérdések?