

SWI-Prolog: XPCE, XML, ODBC

Tomasz Nowak

Kraków, 16 stycznia 2007

Plan prezentacji

- 1 **XPCE**
 - Wprowadzenie do XPCE
 - Charakterystyka XPCE
 - Pomocne narzędzia
- 2 **XML**
 - SWI-Prolog SGML/XML parser
 - Parsowanie dokumentu XML
 - Lista predykatów
- 3 **ODBC**
 - Wstęp
 - Połączenie Prologa z PostgreSQL
 - Podstawowe predykaty

Plan prezentacji

1 XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- Pomocne narzędzia

2 XML

- SWI-Prolog SGML/XML parser
- Parsowanie dokumentu XML
- Lista predykatów

3 ODBC

- Wstęp
- Połączenie Prologa z PostgreSQL
- Podstawowe predykaty

Co to jest XPCE? I

XPCE

Pakiet narzędziowy dla tworzenia graficznych aplikacji w Prologu i innych interakcyjnych i dynamicznych typach języków. Specyficzna biblioteka która dostarcza wszelkich mechanizmów obiektowo zorientowanym językom programowania, takich jak: klasy, obiekty, metody, dziedziczenie, warunki, iteracje itp., co umożliwia tworzenie własnych komponentów.

Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- **Charakterystyka XPCE**
- Pomocne narzędzia

2

XML

- SWI-Prolog SGML/XML parser
- Parsowanie dokumentu XML
- Lista predykatów

3

ODBC

- Wstęp
- Połączenie Prologa z PostgreSQL
- Podstawowe predykaty

Elementy XPCE

- Predefiniowane klasy i elementy wchodzące w skład pakietu:
 - Okienka
 - Obiekty graficzne (linie, okręgi, tekst, bitmapy, itp.)
 - Elementy menu (przyciski, różne rodzaje menu (*pull-down/popup*), itp.)
 - Reprezentacje danych (tablice, łańcuchy, itp.)
 - Programowanie PCE (klasy, metody, itp.)

Właściwości I

- Cechy XPCE, które sprawiają że jest to wygodne narzędzie do szybkiego tworzenia graficznego interfejsu przenośnego pomiędzy systemami operacyjnymi UNIX i Windows:
 - Dodana warstwa obiektowa do Prologu
 - Wbudowana grafika jest zdefiniowana w języku C dla uzyskania szybkości
 - Metody mogą otrzymywać argumenty jako dane Prologa, natomiast dane Prologa mogą być kojarzone ze zmiennymi XPCE
 - Wysoki poziom abstrakcji warstwy graficznej

Właściwości II

- Wykorzystanie szybkiego cyklu projektowania Prologu
- Wirtualna maszyna XPCE (*VWS – Virtual Windows System*) oraz wbudowana biblioteka klas jest napisana w standardowym ANSI-C i jest przenośna do każdej maszyny oferującej odpowiedni model pamięci (32 lub 64 bitowy).

Podstawowe predykaty

new(*?Reference, +TermDescription*)

Powyższy predykat tworzy obiekt i albo przydziela referencję do niego albo łączy pierwszy argument z referencją wygenerowaną przez XPCE.

?- *new (@p, point(10,10)).*

send(*+Receiver, +Selector(+Argument...)*)

Predykat ten manipuluje stanem obiektu. Pierwszy argument predykatu jest referencją obiektu, drugi jest termem, który jest nazwą metody do wywołania z podanymi argumentami operacji do wykonania.

?- *send(@ci, fill_pattern, colour(orange)).*

Podstawowe predykaty

get (+Receiver, +Selector(+Argument...), -Result)

Pobiera stan obiektu. Pierwsze dwa argumenty są takie same jak dla predykatu *send* natomiast ostatni jest związany ze zwracaną wartością. Zwracane wartości są referencją z wyjątkiem obiektów *name*, *integer* *real*.

?- *get(@display,size,Size)*.

free(+Reference)

Predykat ten manipuluje stanem obiektu. Pierwszy argument predykatu jest referencją obiektu, drugi jest termem, który jest nazwą metody do wywołania z podanymi argumentami operacji do wykonania.

?- *free(@display)*.

Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- **Pomocne narzędzia**

2

XML

- SWI-Prolog SGML/XML parser
- Parsowanie dokumentu XML
- Lista predykatów

3

ODBC

- Wstęp
- Połączenie Prologa z PostgreSQL
- Podstawowe predykaty

Narzędzia XPCE

Z powodu dużego rozmiaru biblioteki XPCE ze środowiskiem został zintegrowany rozbudowany system pomocy (uruchomienie za pomocą predykatu *manpce.*). System ten zawiera dużą liczbę narzędzi umożliwiającą analizowanie różnych aspektów środowiska.

Lista narzędzi

- **Browsers/Class Hierarchy** – umożliwia użytkownikowi analizę hierarchii klas XPCE, pokazuje relacje dziedziczenia.
- **Browsers/Class Browser** – Najważniejsze narzędzie, dostarcza możliwość dostępu do wszystkich elementów XPCE z wyjątkiem interfejsu Prologa.
- **Browsers/Search** – przeszukiwanie zawartości manuala.
- **Browsers/Global Objects** – narzędzie pozwalające znaleźć predefiniowane obiekty XPCE (@pce, @prolog, @arg1, itp.).

Lista narzędzi

- Browsers/Class Hierarchy – umożliwia użytkownikowi analizę hierarchii klas XPCE, pokazuje relacje dziedziczenia.
- Browsers/Class Browser – Najważniejsze narzędzie, dostarcza możliwość dostępu do wszystkich elementów XPCE z wyjątkiem interfejsu Prologa.
- Browsers/Search – przeszukiwanie zawartości manuala.
- Browsers/Global Objects – narzędzie pozwalające znaleźć predefiniowane obiekty XPCE (@pce, @prolog, @arg1, itp.).

Lista narzędzi

- Browsers/Class Hierarchy – umożliwia użytkownikowi analizę hierarchii klas XPCE, pokazuje relacje dziedziczenia.
- Browsers/Class Browser – Najważniejsze narzędzie, dostarcza możliwość dostępu do wszystkich elementów XPCE z wyjątkiem interfejsu Prologa.
- Browsers/Search – przeszukiwanie zawartości manuala.
- Browsers/Global Objects – narzędzie pozwalające znaleźć predefiniowane obiekty XPCE (@pce, @prolog, @arg1, itp.).

Lista narzędzi

- Browsers/Class Hierarchy – umożliwia użytkownikowi analizę hierarchii klas XPCE, pokazuje relacje dziedziczenia.
- Browsers/Class Browser – Najważniejsze narzędzie, dostarcza możliwość dostępu do wszystkich elementów XPCE z wyjątkiem interfejsu Prologa.
- Browsers/Search – przeszukiwanie zawartości manuala.
- Browsers/Global Objects – narzędzie pozwalające znaleźć predefiniowane obiekty XPCE (@pce, @prolog, @arg1, itp.).

Lista narzędzi

- **Browsers/XPCE/Prolog Predicates – dokumentacja wszystkich dostępnych predykatów**
- Tools/Visual Hierarchy – umożliwia obejrzenie struktury interfejsu użytkownika, pomocne np. przy znajdowaniu referencji do obiektów.
- File/Demo Programs – przegląd dostępnych przykładowych programów, możliwe uruchomienie i przeglądanie źródeł.

Lista narzędzi

- Browsers/XPCE/Prolog Predicates – dokumentacja wszystkich dostępnych predykatów
- Tools/Visual Hierarchy – umożliwia obejrzenie struktury interfejsu użytkownika, pomocne np. przy znajdowaniu referencji do obiektów.
- File/Demo Programs – przegląd dostępnych przykładowych programów, możliwe uruchomienie i przeglądanie źródeł.

Lista narzędzi

- Browsers/XPCE/Prolog Predicates – dokumentacja wszystkich dostępnych predykatów
- Tools/Visual Hierarchy – umożliwia obejrzenie struktury interfejsu użytkownika, pomocne np. przy znajdowaniu referencji do obiektów.
- File/Demo Programs – przegląd dostępnych przykładowych programów, możliwe uruchomienie i przeglądanie źródeł.

Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- Pomocne narzędzia

2

XML

- **SWI-Prolog SGML/XML parser**
- Parsowanie dokumentu XML
- Lista predykatów

3

ODBC

- Wstęp
- Połączenie Prologa z PostgreSQL
- Podstawowe predykaty

Pakiet sgml2pl

To biblioteka napisana w C do parsowania dokumentów SGML i XML, która zwraca zarówno informację o dokumencie jak i o DTD(ang. Document Type Definition) dokumentu. Cechy parsera sgml2pl:

- Mały rozmiar (mniej niż 100kB programu wykonywalnego)
- Szybkość
- Dostęp do DTD
- Elastyczna obsługa wejścia

Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- Pomocne narzędzia

2

XML

- SWI-Prolog SGML/XML parser
- **Parsowanie dokumentu XML**
- Lista predykatów

3

ODBC

- Wstęp
- Połączenie Prologa z PostgreSQL
- Podstawowe predykaty

Przykład parsowania pliku XML/HTML

Plik wejściowy

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">  
  
<html>  
<head>  
<title>Demo</title>  
</head>  
<body>  
<h1 align=center>This is a demo</title>  
<p>Paragraphs in HTML need not be closed.  
</body>  
</html>
```

Zapytanie

```
?- load_html_file('plik.html', Term),  
   pretty_print(Term).
```

Przykład parsowania pliku XML/HTML

Wyjście

```
[ element (html,  
  [ version = '-//W3C//DTD HTML 4.0 Transitional//EN'  
  ],  
  [ element (head,  
    [],  
    [ element (title,  
      [],  
      [ 'Demo'  
    ])  
  ],  
  element (body,  
    [],  
    [ element (h1,  
      [ align = center  
      ],  
      [ 'This is a demo',  
        element (p,  
          [],  
          [ 'Paragraphs in HTML need not be closed.'  
        ])  
      ])  
    ])  
  ])  
].
```


Przykład parsowania pliku XML/HTML

Ostrzeżenia parsera

```
Warning: SGML2PL(sgml): plik.html:9: Ignored end-tag for "title" which is not open  
Warning: SGML2PL(sgml): plik.html:10: Element "p" not allowed here  
Warning: SGML2PL(sgml): plik.html:11: Inserted omitted end-tag for "h1"
```

- Przetworzony przez parser dokument jest reprezentowany jako lista. Atomy to CDATA a termy w postaci element(Nazwa, Atrybuty, Zawartość) to kolejne elementy dokumentu.

Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- Pomocne narzędzia

2

XML

- SWI-Prolog SGML/XML parser
- Parsowanie dokumentu XML
- **Lista predykatów**

3

ODBC

- Wstęp
- Połączenie Prologa z PostgreSQL
- Podstawowe predykaty

Lista wybranych predykatów

- *html_write(+Stream, +Term, +Options)*
Tworzy dokument HTML z odpowiednio przygotowanego termu
- *load_html_file(+File, -Content)*
Parsuje dokument HTML do postaci termu w prologu
- *load_structure(+Source, -ListOfContent, +Options)*
Parsuje dane XML/SGML/HTML do postaci termu w prologu
- *load_xml_file(+File, -ListOfContent)*
Parsuje plik XML do postaci termu w prologu
- *get_sgml_parser(+Parser, -Option)*
Pobiera opcje parsera

Lista wybranych predykatów

- *html_write(+Stream, +Term, +Options)*
Tworzy dokument HTML z odpowiednio przygotowanego termu
- *load_html_file(+File, -Content)*
Parsuje dokument HTML do postaci termu w prologu
- *load_structure(+Source, -ListOfContent, +Options)*
Parsuje dane XML/SGML/HTML do postaci termu w prologu
- *load_xml_file(+File, -ListOfContent)*
Parsuje plik XML do postaci termu w prologu
- *get_sgml_parser(+Parser, -Option)*
Pobiera opcje parsera

Lista wybranych predykatów

- *html_write(+Stream, +Term, +Options)*
Tworzy dokument HTML z odpowiednio przygotowanego termu
- *load_html_file(+File, -Content)*
Parsuje dokument HTML do postaci termu w prologu
- ***load_structure(+Source, -ListOfContent, +Options)***
Parsuje dane XML/SGML/HTML do postaci termu w prologu
- *load_xml_file(+File, -ListOfContent)*
Parsuje plik XML do postaci termu w prologu
- *get_sgml_parser(+Parser, -Option)*
Pobiera opcje parsera

Lista wybranych predykatów

- *html_write(+Stream, +Term, +Options)*
Tworzy dokument HTML z odpowiednio przygotowanego termu
- *load_html_file(+File, -Content)*
Parsuje dokument HTML do postaci termu w prologu
- *load_structure(+Source, -ListOfContent, +Options)*
Parsuje dane XML/SGML/HTML do postaci termu w prologu
- *load_xml_file(+File, -ListOfContent)*
Parsuje plik XML do postaci termu w prologu
- *get_sgml_parser(+Parser, -Option)*
Pobiera opcje parsera

Lista wybranych predykatów

- *html_write(+Stream, +Term, +Options)*
Tworzy dokument HTML z odpowiednio przygotowanego termu
- *load_html_file(+File, -Content)*
Parsuje dokument HTML do postaci termu w prologu
- *load_structure(+Source, -ListOfContent, +Options)*
Parsuje dane XML/SGML/HTML do postaci termu w prologu
- *load_xml_file(+File, -ListOfContent)*
Parsuje plik XML do postaci termu w prologu
- *get_sgml_parser(+Parser, -Option)*
Pobiera opcje parsera

Lista wybranych predykatów

- *new_sgml_parser(-Parser, +Options)*
Tworzy nowy parser
- *set_sgml_parser(+Parser, +Option)*
Ustawia opcje parsera(typ dokumentu, kodowanie, itp.)
- *sgml_parse(+Parser, +Options)*
Parsuje strumień podany w opcjach
- *xml_write(+Stream, +Term, +Options)*
Tworzy plik XML z odpowiednio przygotowanego termu

Lista wybranych predykatów

- *new_sgml_parser(-Parser, +Options)*
Tworzy nowy parser
- *set_sgml_parser(+Parser, +Option)*
Ustawia opcje parsera (typ dokumentu, kodowanie, itp.)
- *sgml_parse(+Parser, +Options)*
Parsuje strumień podany w opcjach
- *xml_write(+Stream, +Term, +Options)*
Tworzy plik XML z odpowiednio przygotowanego termu

Lista wybranych predykatów

- *new_sgml_parser(-Parser, +Options)*
Tworzy nowy parser
- *set_sgml_parser(+Parser, +Option)*
Ustawia opcje parsera(typ dokumentu, kodowanie, itp.)
- ***sgml_parse(+Parser, +Options)***
Parsuje strumień podany w opcjach
- *xml_write(+Stream, +Term, +Options)*
Tworzy plik XML z odpowiednio przygotowanego termu

Lista wybranych predykatów

- *new_sgml_parser(-Parser, +Options)*
Tworzy nowy parser
- *set_sgml_parser(+Parser, +Option)*
Ustawia opcje parsera (typ dokumentu, kodowanie, itp.)
- *sgml_parse(+Parser, +Options)*
Parsuje strumień podany w opcjach
- ***xml_write(+Stream, +Term, +Options)***
Tworzy plik XML z odpowiednio przygotowanego termu

Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- Pomocne narzędzia

2

XML

- SWI-Prolog SGML/XML parser
- Parsowanie dokumentu XML
- Lista predykatów

3

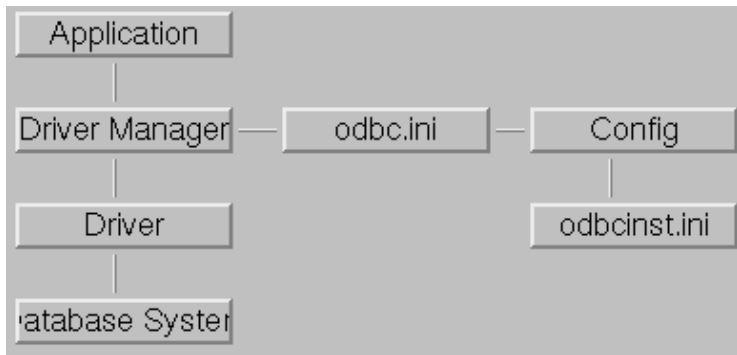
ODBC

- **Wstęp**
- Połączenie Prologa z PostgreSQL
- Podstawowe predykaty

ODBC

Open DataBase Connectivity

Otwarte łącze baz danych czyli interfejs pozwalający programom łączyć się z systemami zarządzającymi bazami.



Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- Pomocne narzędzia

2

XML

- SWI-Prolog SGML/XML parser
- Parsowanie dokumentu XML
- Lista predykatów

3

ODBC

- Wstęp
- **Połączenie Prologa z PostgreSQL**
- Podstawowe predykaty

Wykorzystane elementy

- DataBase System – PostgreSQL 8.1.5
- ODBC Driver – odbc-postgresql
- Driver Manager – unixODBC
- Application – Swi-Prolog z pakietem odbc

Plan prezentacji

1

XPCE

- Wprowadzenie do XPCE
- Charakterystyka XPCE
- Pomocne narzędzia

2

XML

- SWI-Prolog SGML/XML parser
- Parsowanie dokumentu XML
- Lista predykatów

3

ODBC

- Wstęp
- Połączenie Prologa z PostgreSQL
- **Podstawowe predykaty**

Zarządzanie połączeniem

- *odbc_connect(+DSN, -Connection, +Options)*

```
open_wordnet :-
    odbc_connect('WordNet', _,
        [ user(jan),
          password(xxx),
          alias(wordnet),
          open(once)
        ]).
```

Zarządzanie połączeniem

- `odbc_disconnect(+Connection)`
- `odbc_set_connection(+Connection, +Option)`
- `odbc_get_connection(+Connection, ?Property)`

Wykonywanie zapytań SQL

- *odbc_query(+Connection, +SQL, -RowOrAffected, +Options)*

```
lemma(Lemma) :-  
    odbc_query(wordnet,  
                'SELECT (lemma) FROM word',  
                row(Lemma)).
```

```
insert_child(Child, Mother, Father, Affected) :-  
    odbc_query(parents,  
                'INSERT INTO parents (name,mother,father)  
                VALUES ("mary", "christine", "bob")',  
                affected(Affected)).
```

Sparametryzowane zapytania SQL

- *odbc_prepare(+Connection, +SQL, +Parameters, -Statement, +Options)*

```
odbc_prepare(baza,  
  'SELECT (wname) FROM word WHERE wname like ?',  
  [varchar(64)  
  ],  
  Qid),
```

- *odbc_execute(+Statement, +ParameterValues, -RowOrAffected)*

Informacje dodatkowe

- Prezentacja dotyczyła programów w wersji:
 - SWI-Prolog (Multi-threaded, Version 5.6.25)
 - XPCE 6.6.25, December 2006 for i686-linux and X11R6
 - SWI-Prolog SGML/XML parser Version 2.0.5, March 2005
 - unixODBC 2.2.11
- Prezentacja została stworzona w oparciu o klasę beamer.cls w wersji 3.06

Przydatne linki I



<http://www.swi-prolog.org/>

SWI-Prolog Reference Manual



[http://www.swi.psy.uva.nl/projects/xpce/
UserGuide](http://www.swi.psy.uva.nl/projects/xpce/UserGuide)

Programming in XPCE/Prolog



[http:
//www.swi-prolog.org/packages/sgml2pl.html](http://www.swi-prolog.org/packages/sgml2pl.html)

SWI-Prolog SGML/XML parser

Przydatne linki II



`http://www.swi-prolog.org/packages/odbc.html`

SWI-Prolog ODBC Interface



`http://www.unixodbc.org/doc/UserManual/`

unixODBC USER MANUAL

Koniec

Dziękuję za ewentualną uwagę.