



Webprogramozás Prologban

Hunyadi Levente

2005. október 3.

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web

Container és

Prolog Server

Pages

Programming in Logic Languages on the Web

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web

Container és

Prolog Server

Pages

- HTML és XML szerkezet Prolog kifejezésként történő ábrázolása
- egyszerű HTML és XML sablonok kezelése
- átlátszó működés CGI környezetben
- egyszerű HTTP 1.0 kérés protokoll megvalósítás

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web

Container és

Prolog Server

Pages

Kanonikus alak:

A PiLLoW *html2terms/2* predikátuma a karakterlistaként megadott dokumentumot kanonikus Prolog HTML kifejezéssé alakítja.

Az *env*(címké, attribútumlista, tartalomlista) kifejezés szolgál tulajdonságokkal felruházott, tartalommal bíró HTML elemek (*element*) ábrázolására, pl.

```
<a href="http://www.cs.bme.hu/">cs.bme.hu</a>
```

```
[env(a, [href="http://www.cs.bme.hu/"],  
["cs.bme.hu"])]
```

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

A $\$$ (címké, attribútumok) alak üres elemek ábrázolását teszi lehetővé, pl.

```

```

```
[img $ [src="image.png"]]
```

A $\$/2$ -t a PiLLoW bináris operátorként definiálja.

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

a *comment*(füzér) tárolja a HTML forrásban elhelyezett megjegyzéseket

a *declare*(füzér) a dokumentum elején lévő dokumentumtípus-deklarációt (DOCTYPE) tartalmazza minden egyéb, címkéken kívül előforduló szöveget karakterlisták (füzérek) tárolnak

[PiLLoW](#)

[Áttekintés](#)

[HTML és XML](#)

[Sablonok](#)

[CGI](#)

[Prolog Web
Container és
Prolog Server
Pages](#)

A *html2terms/2* predikátumhoz hasonló működés váltható ki az *xml2terms/2* eljárás hívásával.

A két formátum közti eltérésből fakadó különbségek:

- XML-ben lezáratlan címkék nem fordulhatnak elő. `<a>` tömör írásmódját a PiLLoW az *elem*(címké, attribútumlista) kifejezéssel ábrázolja, a *\$/2* nem használható.
- *xmldecl*(attribútumlista) a dokumentum elején álló `<?xml attribútumlista ?>` struktúrának felel meg.

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

A *html2terms/2* és *xml2terms/2* predikátumok ellenirányban is használhatóak, amennyiben első argumentumuk változó. Ekkor a programozó további kényelmi kifejezésekkel élhet:

- *image*(forráshivatkozás) és *image*(forráshivatkozás, attribútumlista) kép beillesztésére,
- *ref*(célhivatkozás, szöveg) hivatkozás létrehozásához,
- *heading*(szint, szöveg) fejezetcím beszúrásához, a `h1`, `h2`, ... `h6` HTML elemeknek felel meg,

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

- *itemize*(elemlista) sorszámozatlan lista készítéséhez,
- *enumerate*(elemlista) sorszámozott lista készítéséhez,
- *verbatim*(szöveg) olyan szöveg beszúrásához, amelyben HTML-ben különleges jelentéssel bíró karakterek szöktető-sorozatukkal (escape sequence) szerepelnek.

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web

Container és

Prolog Server

Pages

Példa kifejezés HTML dokumentummá alakítására

Prolog kifejezés:

```
env(html, [], [ env(body, [lang=hu],  
[ heading(1, title), text,  
ref('http://dp.iit.bme.hu', hyperlink) ]) ])
```

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

Példa kifejezés HTML dokumentummá alakítására

Létrehívott dokumentumszöveg (emberi formában):

```
<html> <body lang="hu"> <h1>title</h1> text  
<a ref="http://dp.iit.bme.hu">hyperlink</a>  
</body> </html>
```

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web

Container és

Prolog Server

Pages

Gyakran szükség van arra, hogy HTML oldalakat paraméterezzünk, azaz bizonyos részleteit dinamikusan hozzuk létre. Ehhez nyújt segítséget a *html_template/3* predikátum.

[PiLLoW](#)

[Áttekintés](#)

[HTML és XML](#)

[Sablonok](#)

[CGI](#)

[Prolog Web
Container és
Prolog Server
Pages](#)

- Első argumentuma egy előre megformázott forrásállományt képviselő karakterlista. A forrásállományban különleges jelölést alkalmazunk: az aláhúzásjellel kezdődő attribútumértékek és a `v` címkék dinamikusan illesztett elemek helyét jelölik.
- Második argumentuma változó, amely a létrehozott Prolog kifejezést tartalmazza majd. A kifejezés nem lesz tömör, hanem a forrásállományban jelölt attribútumok értékeként, illetve a HTML-fa `v` címkéjű ágai helyett változók állnak majd.
- A harmadik argumentumot a predikátum egy szótárhoz köti. Itt `Név=Változó` alakban jelennek meg a forrásállományban nevesített változók.

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

Sablon-forrásállomány

```
<html> <body lang="_language"> <v>content</v>  
</body> </html>
```

[PiLLoW](#)

[Áttekintés](#)

[HTML és XML](#)

[Sablonok](#)

[CGI](#)

[Prolog Web
Container és
Prolog Server
Pages](#)

A *html_template/3* predikátum által létrehozott Prolog kifejezés:

```
env(html, [], [ env(body, [lang=A], [B]) ])
```

A listában visszaadott változómegfeleltetés:

```
[lang=A, content=B]
```

[PiLLoW](#)

[Áttekintés](#)

[HTML és XML](#)

[Sablonok](#)

[CGI](#)

[Prolog Web
Container és
Prolog Server
Pages](#)

Kimeneti állomány

A *memberchk/2* predikátummal a változókat lekérdezve, majd az $A=hu$, illetve $B=ref('http://dp.iit.bme.hu', \text{hyperlink})$ egyesítéseket elvégezve, végül a *html2terms/2* predikátum hívásával az alábbi dokumentumhoz jutunk:

```
<html> <body lang="hu"> <a  
ref="http://dp.iit.bme.hu">hyperlink</a>  
</body> </html>
```

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

Programozási nyelv-független felület kérések végrehajtására. Filozófiája:

- A webkiszolgáló fogadja a böngészőtől érkező HTTP kérést, majd azt az abból származó adatokat két részre osztva feldolgozza, és meghívja a kiszolgáló CGI alkalmazást: a HTTP kérés fejlécéből származó adatok környezeti változókba kerülnek, míg a HTTP kérés törzse az alkalmazás szabványos bemenetén jelenik meg.
- A CGI alkalmazás elemzi a részére átadott adatokat, azok alapján számításokat végez, majd szabványos kimenetén választ generál.
- A webkiszolgáló a CGI alkalmazás választát HTTP válaszba csomagolja.

[PiLLoW](#)

[Áttekintés](#)

[HTML és XML](#)

[Sablonok](#)

[CGI](#)

[Prolog Web
Container és
Prolog Server
Pages](#)

A felület legfontosabb hátránya, hogy különféle CGI alkalmazásoknak bizonyos szabványos teendőket mindenképpen implementálniuk kell. Egy HTML űrlap például GET és POST kérés formájában egyaránt eljuthat az alkalmazáshoz.

- GET esetében a név-érték párok az URL részét képezik, ami a környezeti változók közül nyerhető ki.
- POST esetében a név-érték párokat a szabványos bemenetről olvashatja ki a program

PiLLoW

Áttekintés

HTML és XML

Sablonok

CGI

Prolog Web
Container és
Prolog Server
Pages

Ezeket az eltéréseket célszerű átlátszóvá tenni a program számára. Erre szolgálnak a *get_form_input/1* és *get_form_value/3* predikátumok.

[PiLLoW](#)[Áttekintés](#)[HTML és XML](#)[Sablonok](#)[CGI](#)[Prolog Web
Container és
Prolog Server
Pages](#)

- A `get_form_input/1` a HTTP kérés METHOD paraméterét megvizsgálva eldönti, hogy GET vagy POST kérésről van-e szó. Ettől függően a környezeti változókból vagy a szabványos bemenetről megszerzi az összes név-érték párt, amit egyetlen kimeneti argumentumával, egy ezeket tartalmazó listával, az ún. szótárral egyesít.
- A `get_form_value(szótár, név, érték)` predikátum a szótárból kikeresi a névvel megadott paraméterhez tartozó értéket. Sohasem hiúsul meg, ha a szótárban bejegyzés nem található, nulla hosszúságú atommal tér vissza. A `form_default(érték, alapértelmezés, eredmény)` eljárás szolgál arra, hogy ilyen esetekben alapértelmezett értéket adjunk eredményül. Nemüres érték esetén az eredmény megegyezik az értékkel.

[PiLLoW](#)[Áttekintés](#)[HTML és XML](#)[Sablonok](#)[CGI](#)[Prolog Web
Container és
Prolog Server
Pages](#)

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

Összehasonlítás a PiLLoW lehetőségeivel

A PiLLoW számos előnyös tulajdonsága megkönnyíti a webfelülettel is rendelkező program fejlesztését:

- a HTTP protokoll-specifikus kommunikációt elrejtí,
- sablontechnológiát nyújt a változó tartalom számára,
- már létező, adott célsorozatra választ adó, nem interaktív alkalmazások kis változtatással jeleníthetőek meg weblapon,
- a CGI felület révén egy PiLLoW-ra támaszkodó megoldás bármilyen webkiszolgálóhoz gyorsan illeszthető.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás

Prolog Web
Container
Perzisztencia
Terheléelosztás
Hosszú életű
adatok
Prolog Server
Pages
Lokális változók
Kifejezésnyelv
Címkék
Összefoglalás

Miért nem elegendő a PiLLoW?

Sajnos összetettebb alkalmazások esetén a PiLLoW eszköztára nem elégséges:

- a PiLLoW által nyújtott sablontechnológia csak egyszerű helyettesítéseket tesz lehetővé. Nincs közvetlen mód címkék dinamikus létrehozására, ágak feltételes beillesztésére vagy iterációra. Ugyan ezek megvalósíthatóak Prolog és PiLLoW eszközökkel, ám ez többnyire nehézkes. Mivel gyakran előforduló mintákról van szó, célszerű lenne a lépéseket automatizálni, elrejtetni.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás

Prolog Web
Container
Perzisztencia
Terheléelosztás
Hosszú életű
adatok
Prolog Server
Pages
Lokális változók
Kifejezésnyelv
Címkék
Összefoglalás

Miért nem elegendő a PiLLoW?

- a PiLLoW nem támogatja a megjelenítés és az ún. üzleti logika szétválasztását: a megjelenítési réteg nem független a megoldást szolgáltató Prolog modultól. Másképpen: hiányzik az objektum-orientált nyelvek Model 2 (MVC) szemlélete.
- nincs beépített lehetőség adatok egy kérésen való átörökítésére, a PiLLoW csak egyetlen kérésre adott válasz lebonyolítását segíti.
- a CGI architektúra nem erőforrás-hatékony: minden kérésre új Prolog folyamat létrehozása szükséges, ezért rosszul skálázható.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás

Prolog Web
Container
Perzisztencia
Terheléelosztás
Hosszú életű
adatok
Prolog Server
Pages
Lokális változók
Kifejezésnyelv
Címkék
Összefoglalás

Példa gyakran használt mintára

PiLLOW segítségével HTML kódrészlet iterálásához

- az elemeket a *findall/3* predikátum segítségével listába gyűjtjük,
- egy PiLLOW HTML sablont olvasunk be egy külső állományból,
- a sablonrészletet a lista elemei szerint különféleképpen helyettesítjük, így HTML fa-részleteket kapunk,
- a részletek listáját egy újabb PiLLOW szerkezetbe ágyazzuk,
- amit karaktersorozattá alakítunk.

PiLLOW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás

Prolog Web
Container
Perzisztencia
Terheléelosztás
Hosszú életű
adatok
Prolog Server
Pages
Lokális változók
Kifejezésnyelv
Címkék
Összefoglalás

- *Perzisztens architektúra.* A böngészőtől érkező kéréseket egy hosszú életű folyamat szolgálja ki, amely folyamatosan fut, ezért nincs szükség újraindulásra minden kérés alkalmával. Az indítás költsége minden program esetén jelentős, megtakarítása számottevő nyereség.
- *Terheléelosztás.* Egyszerre érkező kérések kiszolgálását különböző szálak, illetve (egyszálú Prolog megvalósítások esetén) folyamatok végzik. Egy hosszú időt igénylő (esetleg időkorlátot túllépő) kérés kiszolgálása így nem akasztja meg más felhasználók munkáját.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás

Prolog Web
Container

Perzisztencia

Terheléelosztás

Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

- *Átlátszóság.* A Prolog megvalósítás elől rejtve marad minden HTTP-specifikus elem, a környezet dinamikus tényállításokon keresztül, mint *http_get/2* vagy *http_post/2* elérhető el.
- *Hosszú életű adatok.* Az architektúra támogatja a viszony- (session) és alkalmazásváltozók *assert/1* és *retract/1*-elven történő használatát. A viszonykezelés automatikus.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás

Prolog Web
Container

Perzisztencia

Terheléelosztás

Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A CGI protokoll a programindítási környezetet adatátadásra használja. Egy hosszú életű alkalmazás nem indul újra minden kérés alkalmával, ezért ezeket az adatokat más módon kell eljuttatni a Prolog oldalra. Erre szolgál a socket-alapon megvalósított FastCGI protokoll, amely lehetővé teszi azt is, hogy a webkiszolgáló és a web container különböző gépen fusson.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléelosztás
Hosszú életű
adatok
Prolog Server
Pages

Lokális változók
Kifejezésnyelv
Címkék

Összefoglalás

A beérkező kérések kiszolgálását az éppen szabad szálak (folyamatok) valamelyike végzi el. A feladat elvégzése után a szál visszakerül az ún. thread pool-ba. Ezt a megközelítést több más architektúra, köztük az Apache 2 (worker thread model) és a Microsoft .NET is alkalmazza. Holtidőben a szabad szálak egy FIFO csatorna kimeneténél várakoznak. Új kérés esetén az ún. termelő a feladatot behelyezi a csatornába, amely egy versenyhelyzetet követően valamelyik szabad szálhoz rendelődik hozzá. Lehetőség van a futó szálak számának terheléstől függő változtatására.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container
Perzisztencia

Terheléselosztás

Hosszú életű
adatok
Prolog Server
Pages

Lokális változók
Kifejezésnyelv
Címkék
Összefoglalás

A közös erőforrások elérése szabályozást igényel. Egy kiragadott példa a Prolog kiszolgálóoldali lapokhoz (PSP) történő hozzáférés: a gyors kiszolgálás érdekében első hivatkozásukkor a rendszer beolvassa és értelmezi őket. Lényeges, hogy más szál ne kezdje meg az értelmezést, ha az már folyamatban van, hanem várjon annak befejeztére és a befejezésről valamilyen üzenetprimitívvel értesüljön.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container
Perzisztencia

Terheléselosztás

Hosszú életű
adatok
Prolog Server
Pages

Lokális változók
Kifejezésnyelv
Címkék
Összefoglalás

Szálak helyett folyamatokkal működő rendszer esetén az adatmegosztás nehezebb. A probléma egy egyszálú adatszolgáltató folyamat bevezetésével oldható meg, amely a hozzá érkező kéréseket implicite sorosítja (Linda). A dokumentum-gyorsítótárazást azonban ekkor minden folyamatnak önállóan kell alkalmaznia, így egy értelmezett dokumentum több példányban is jelen lehet.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container
Perzisztencia

Terheléselosztás

Hosszú életű
adatok
Prolog Server
Pages

Lokális változók
Kifejezésnyelv
Címkék

Összefoglalás

Alkalmazásváltozók értékének módosításához szinkronizáció szükséges. A szinkronizáció kritikus szakaszok (más néven *mutual exclusion device*) segítségével, a programozó számára rejtett módon valósul meg az *application_set/2*, *application_update/3* és *application_remove/1* predikátumokon keresztül. Az *application_update/2* lehetőséget biztosít tetszőleges predikátum meghívására, amely a régi érték alapján meghatározza az újat, mindezt a zárok felengedése nélkül, amíg a frissítés tart:

```
updater_predicate(OldValue, NewValue) :-  
  NewValue is OldValue + 1.  
psp:application_update(counter,  
  updater_predicate).
```

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléelosztás

Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

Viszonyváltozók értékének átlátszó módosítása kevésbé összetett feladat. Egy kérés kiszolgálását három részre osztjuk: prolókus, törzs, epilógus. A prolókus a szál dinamikus tényállításaiként a programhoz illeszti a kérés környezetre vonatkozó adatokat, illetve a több szál által közösen használt területen elhelyezkedő hosszú életű (viszony) adatokat. A törzs a programozó által meghatározott rész, amely a dinamikus tényállításokat tetszőlegesen módosíthatja. Az epilógus a megváltozott tényállítás-halmaz vonatkozó részét visszahelyezi a közösen használt területre.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléelosztás

**Hosszú életű
adatok**

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A Prolog Server Pages tulajdonságai

A Prolog Server Pages

- XML 1.0 szintaxisú
- XSLT-hez hasonló deklaratív nyelv,
- amely beépítetten támogatja a feltételes beillesztést, az iterációt, a lokális változókötést és elemek dinamikus létrehozását;
- új, szabadon definiált címkékkel bővíthető;
- attribútumok értékeiként lehetővé teszi kifejezésnyelvi elemek használatát;
- adatok újrafelhasználását segíti változókon keresztül;
- megvalósítja a megjelenítés és az üzleti logika teljes szétválasztását.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

Az XSLT mintájára a Prolog Server Pages is lehetővé teszi változók használatát. A Prolog szemantikához hasonlóan változók csak egyszer kaphatnak értéket. Minden változó lokális a környezetére, azaz a változóértékdadás (a deklaráció és értékdadás egyszerre történik) a szülőelem deklaráció utáni gyermekeire terjed ki, tetszőleges mélységig. Az azonos nevű változók közül a későbbi elfedi a korábbi.

```
<a><!-- itt még nem használható a változó ->
<psp:capture var="local">lokális változó
értéke</psp:capture> <!-- már hivatkozhatunk
a változóra -> <b> <!-- a láthatóság a szülő
további gyermekelemeire is kiterjed -> </b>
</a>
```

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléelosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A Prolog Server Pages attribútumaiban { és } jelek között típusos kifejezésnyelv használható. Megengedett a legtöbb, *is/2* jobb oldalán álló aritmetikai kifejezés, illetve több füzérkezelő függvény használata, de definiálhatunk új függvényeket is. A kivételt okozó kifejezések (pl. osztás nullával, szöveg és szám összeadása, stb.) a *null* típusba tartozó `null` atomot veszik fel, amelynek jelentése megfelel a háromértékű logikákban megszokottnak és nem azonos a hamis konstanssal.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A speciális funkcionalitást megvalósító címkék a *psp* névtérbe tartoznak. Ilyen címkéjű elem sohasem kerül a kimenetre, hanem attribútumainak és tartalmának megfelelően elemek sorozatára kiértékelődik ki, amelyek az elem helyére kerülnek. A kiértékelés felülről lefelé (top-down) elven zajlik.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

Egyszerűsített példa címke kiértékelésére

```
<psp:for-all predicate="member_of_list([1,2,3])">  
<psp:if expr="{iterator eq 2}"> kettő </psp:if>  
</psp:for-all>
```

```
<psp:if expr="{1 eq 2}"> kettő </psp:if>  
<psp:if expr="{2 eq 2}"> kettő </psp:if>  
<psp:if expr="{3 eq 2}"> kettő </psp:if>
```

kettő

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A `psp:if goal="predicate(A, B, C, ...), ..."` címke beilleszti tartalmát, ha az adott célsorozat bizonyítható, ellenkező esetben üres tartalmat ad.

```
<psp:if goal="member(A, [1,2,3]), A gt 2"> igaz  
</psp:if>
```

A `psp:if expr="expression"` címke beilleszti tartalmát, ha az adott kifejezés igaz (true), ellenkező esetben üres tartalmat ad.

```
<psp:if expr="{10 / 2 gte 5}"> igaz </psp:if>
```

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A *psp:if-else goal="predicate(A, B, C, ...), ..."* címke működése megegyezik a *psp:if* címkéjével, meghiúsulás esetén azonban a *psp:else* tartalmaként megadott részletet adja.

```
<psp:if-else> <psp:if goal="member(A, [1,2,3]),  
A gt 2"> igaz </psp:if> <psp:else> hamis  
</psp:else> </psp:if-else>
```

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A `psp:choose` címke belsejébe ágyazott `psp:when` `goal="predicate(A, B, C, ...), ..."` címkesorozat elágazás-sorozattá alakul.

```
<psp:choose> <psp:when goal="goal1"> ha  
goal1 bizonyítható </psp:when> <psp:when  
goal="goal2"> ha goal2 bizonyítható </psp:when>  
<psp:otherwise> egyébként <psp:otherwise>  
</psp:choose>
```

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléelosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

Többszörös elágazás kifejtése

```
<psp:if-else> <psp:if goal="goal1"> ha goal1
bizonyítható </psp:if> <psp:else> <psp:if-else>
<psp:if goal="goal2"> ha goal2 bizonyítható
</psp:if> <psp:else> egyébként </psp:else>
</psp:if-else> </psp:else> </psp:if-else>
```

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

A *psp:for-each function="predicate(A, B, C, ...)"* a jelölt predikátumot egy további kimeneti argumentummal bővíti, amelyet a predikátum az eredménylistával egyesít. A címke tartalma a lista minden elemével kiértékelődik, a kimeneti argumentum aktuális értékéhez az *iterator* lokális változón keresztül félhetünk hozzá.

A *psp:for-each-else function="predicate(A, B, C, ...)"* hasonló működésű, megghiúsulás esetén azonban az egyébként ág tartalmát illeszti a kimenetbe, hasonlóan a *psp:if* címkéhez.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléelosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók
Kifejezésnyelv

Címkék

Összefoglalás

A *psp:for-all function="predicate(A, B, C, ...)"* nemdeterminisztikus predikátum esetén használatos, a megoldásokat listába gyűjti, majd annak elemein lép végig *psp:for-each* szemantikája szerint.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás

Lokális változó értékadás és lekérdezés

A *psp:capture var="varname"* címke a nyitó és záróelem által közrefogott tartalmat a megnevezett lokális változónak adja értékül.

A *psp:insert var="varname"* címke a megjelölt változó értékét szűri a kimeneti dokumentumba.

A *psp:insert expr="expression"* címke hatására a kifejezés kiértékelődik és eredménye a kimeneti dokumentumba kerül.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók
Kifejezésnyelv

Címkék

Összefoglalás

További (meta)lehetőségek

A *psp:cached* címke használható csak egyszer kiértékelendő tartalom megjelölésére.

A *psp:include src="sourcefile"* üres címkével a hivatkozott dokumentumot illeszthetjük a címke helyére, a kiértékelés úgy folytatódik, mintha a dokumentum tartalmát közvetlenül a címke helyére másoltuk volna.

A *psp:import src="sourcefile"* üres címke működése fenti társáéhoz hasonló, a hivatkozott dokumentum azonban önálló egységet alkot, nem hivatkozhat a forrásdokumentumban definiált változókra, de kapcsolódhat hozzá önálló üzleti logikai kód.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléelosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók
Kifejezésnyelv

Címkék

Összefoglalás

A *Prolog Web Container* a vezérlési feladatokat látja el: átlátszóan biztosítja a hosszú életű folyamatként történő működést és a kérések több szálon történő kiszolgálását.

A *Prolog Server Pages* a megjelenítés és az üzleti logika szétválasztását biztosítja: a Prolog nyelven írt üzleti logika a megjelenítéstől elválasztott modulban található, amelynek egyes predikátumaira a megjelenítést biztosító PSP dokumentumok hivatkozhatnak.

Köszönöm a figyelmet!

A keretrendszer további lehetőségei és megvalósításának részletei a <http://hunyadi.uw.hu/prosper/report.pdf> címen olvashatók.

PiLLoW

Prolog Web
Container és
Prolog Server
Pages

Összehasonlítás
Prolog Web
Container

Perzisztencia

Terheléselosztás
Hosszú életű
adatok

Prolog Server
Pages

Lokális változók

Kifejezésnyelv

Címkék

Összefoglalás