

Wissensbasierte Katalogsysteme

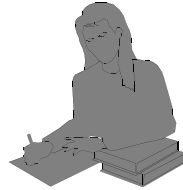
- Ausfüllhilfe
- Empfehlungen
- Konfiguration
- Führung
- Wenig Tastendrucke
- Wenige Seiten
- Wenig Anklickbares auf einer Seite
- Wenig Herumsuchen

Nur gerade die gewünschten Waren
in der gewünschten Ausprägung sehen!

Vergleichen Sie:

<http://design.peugeot.co.uk> <http://mbks2.mercedes-benz.com/>

Architektur



Wünsche

Schnittstelle
(WWW, WAP, PDA)

Konfigurator

Filter

Produkte

Layout
Formulare



Erfassung
Import
Wartung

Minimale Menge
von Fragen an BEN

Teilmenge der (noch)
möglichen Eigenschaften

Teilmenge der Tupel
mit bestimmten Eigenschaften

Constraints

Metadaten zu den Produkttabellen ermöglichen

- Wartung der Produktdaten
- Erstellung weniger Seiten mit wenigen geforderten Angaben
- Erkennen unmöglicher Kombinationen von Anforderungen ohne Zugriff auf die Datenbank

Metadaten

```
%hasAsPart(Instanz, Typ, [Bestandteile])  
hasAsPart(T,sitzecke, [tisch, sofa,sessel]).  
hasAsPart(T,tisch,[bein,flaeche]).  
hasAsPart(T,flaeche,[platte,umleimer]).  
...
```

```
%dimension(Teil,Typ, [Dimensionen])  
dimension(T,tisch,[hoch,breit,tief,material,preis,liefer]).  
dimension(T,flaeche,[hoch,breit,tief,material,farbe,preis,liefer]).  
...
```

```
%range(Dimension, [Range])  
%Range: Zahl oder Aufzaehlung  
range(preis,[DM]).  
range(liefer,[Wochen]).  
range(farbe,[blau,rot,schwarz,natur]).  
range(material,[alu, plastik,eiche,buche]).
```

Constraints

```
% dimension(Teil1,[Relation, Wert],  
            Teil2, dimension2,[Relation,Wert])
```

```
%Teil1 bekannt --> Teil2 berechnen
```

```
%Teil2 bekannt --> Teil1 berechnen
```

```
hoch(tisch,[R,W1],bein,hoch,[R,W2]):-
```

```
    (number(W2),
```

```
    W1 is W2 + 3); %evtl. max Bretthoehe aus Metadaten
```

```
    (number(W1), %bei einem Constraint System ist dies unnoetig
```

```
    W2 is W1 - 3).
```

```
hoch(tisch,[R,W1],flaeche,hoch,[R,W2]):-
```

```
    (number(W2),
```

```
    W1 is W2 + 55); %max Beinhoehe aus Metadaten
```

```
    W2 is W1 - 55.
```

```
hoch(flache,Wert,platte,hoch,Wert). %Unifikation nutzen!
```

Constraint Propagation

```
% wertBerechnen(+Liste der Wuensche)
% fuer jeden Wunsch (constraint) werden die Folgen berechnet
% Seiteneffekt durch Aufruf von werteBestimmen(Typ,dimension,Wert)
wertBerechnen([TEW]):-
    append([T,E], [W], TEW), %in Bestandteile zerlegen
    werteBestimmen(T,E,W).
wertBerechnen([X|Wuensche]):-
    append([T,E], [W], X),
    werteBestimmen(T,E,W),
    wertBerechnen(Wuensche).
%% Transformation Listenelement -->Klausel
werteBestimmen(Typ1,E1,Wert1):-
    Z=.. [E1,Typ1,Wert1,Typ2,E2,Wert2],
    call(Z),      %constraints verwenden
    asserta(wunsch(Typ2,E2,Wert2)), %errechnete constraints eintragen
    fail.
werteBestimmen(Typ,E,Wert).
```

Minimale Fragemenge

```
% fragen(+Teil,+Dimensionen,-Range)
fragen(Typ,[X],[X,W]):-
    (clause(wunsch(Typ,X,W),true),
     nonvar(W));           %von BEN angegeben/berechnet ODER
    werte(X,W).           %aus den Metadaten
                           %muss NICHT gefragt werden.
fragen(Typ,[X|Dimensionen],Range):-
    fragen(Typ,[X],Werte),
    fragen(Typ,Dimensionen>WeitereWerte),
    append(Werte>WeitereWerte,Range).
```