Exploratory search through **P**reference-enriched **F**aceted **S**earch

Yannis Tzitzikas
Ass. Professor, Computer Science Department,
University of Crete and
Affiliated Researcher of FORTH-ICS
Heraklion, Crete, GREECE

This work is described in the following paper

- [J. FI 13] Yannis Tzitzikas and Panagiotis Papadakos. Interactive Exploration of Multidimensional and Hierarchical Information Spaces with Real-Time Preference Elicitation. In *Journal FUNDAMENTA INFORMATICAE*, Volume 122, Issue 4, pp 357-399, 2013.
- The PFS method was the topic of the PhD thesis of Panagiotis Papadakos (he is now post-doctoral researcher in our lab).

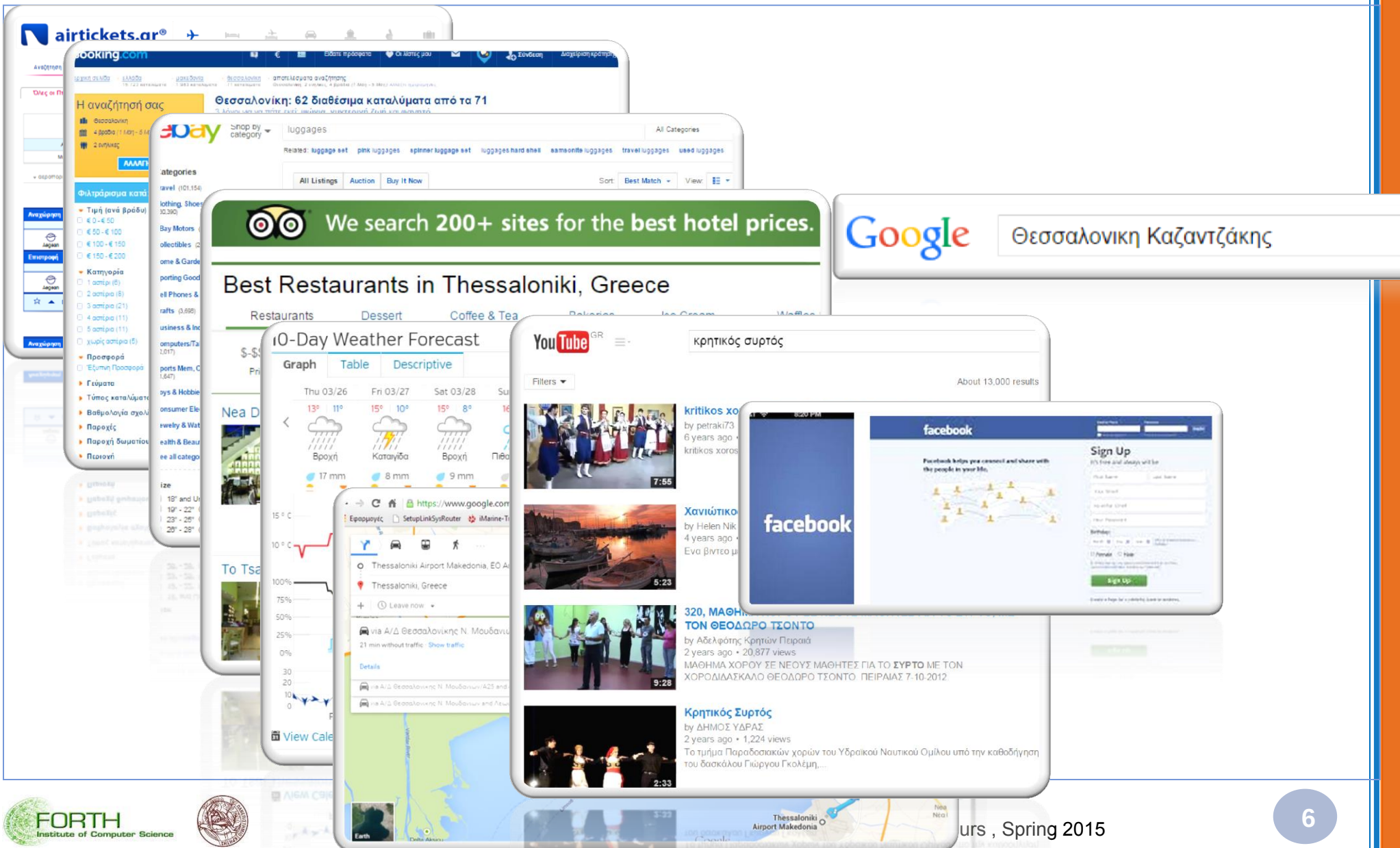
Outline

- Introduction and Background
  - Exploratory Search
  - Faceted Exploration
- Preferences
- Preference-enriched Faceted Search (PFS)
  - Demo Hippalus
  - Formally
    - FS
    - PFS
    - Algs
  - Evaluation
  - Current Investigations
- Concluding Remarks
- References
- Acknowledgements
- [Extras]

Introduction and Background

Exploratory Search

Faceted Exploration


airtickets.gr
booking.com
Θεσσαλονίκη: 62 διαθέσιμα καταλύματα από τα 71
eBay
luggages
We search 200+ sites for the best hotel prices.
Best Restaurants in Thessaloniki, Greece
Restaurants
Dessert
Coffee & Tea
10-Day Weather Forecast
Graph
Table
Descriptive
Google
Θεσσαλονικη Καζαντζάκης
YouTube GR
κρητικός συρτός
Filters
About 13,000 results
kritikos xo
by petraki73
6 years ago
kritikos xoros
7:55
Χανιώτικο
by Helen Nik
4 years ago
Ενα βιντεο μ
5:23
320, ΜΑΘΗ
ΤΟΝ ΘΕΟΔΩΡΟ ΤΣΟΝΤΟ
by Αδελφότης Κρητών Πειραιά
2 years ago • 20,877 views
ΜΑΘΗΜΑ ΧΟΡΟΥ ΣΕ ΝΕΟΥΣ ΜΑΘΗΤΕΣ ΓΙΑ ΤΟ ΣΥΡΤΟ ΜΕ ΤΟΝ ΧΟΡΟΔΙΔΑΣΚΑΛΟ ΘΕΟΔΩΡΟ ΤΣΟΝΤΟ ΠΕΙΡΑΙΑΣ 7-10-2012
9:28
Κρητικός Συρτός
by ΔΗΜΟΣ ΥΔΡΑΣ
2 years ago • 1,224 views
Το τμήμα Παραδοσιακών χορών του Υδραϊκού Ναυτικού Ομίλου υπό την καθοδήγηση του δασκάλου Γιώργου Γκολέμη,...
2:33
facebook
Sign Up
Thessaloniki Airport Makedonia, EO
Thessaloniki, Greece
Leave now
21 min without traffic · Show traffic
Details

What Users usually Want/Do when Searching?

Kinds of information needs

- *Precision-oriented*
  - Locate one resource or/and its attributes

    e.g. Find the telephone of a store, the website of a firm or person
- *Recall-oriented*
  - Locate a **set** of resources

    e.g. Medical information seeking, travel planning, e-shopping

Recall-Oriented Information Needs

- In Recall-Oriented Information Needs:
  - the users require >1 hit
  - essentially such needs correspond to decision tasks
- Examples of Recall-oriented information needs
  - Booking of flights, hotels, …
  - Product-buying
  - Bibliography search
  - Patent Search
  - Medical Search
  - ….

Over 60% of web search queries are ***recall-oriented***
[Broder 02, Rose and Levinson 04]

How many of you have used a system like **booking.com**?

How many of you have booked the 1st suggestion returned by **booking.com** without looking at any of the rest hotels?

How many of you have entirely read the 1st paper that **Scholar Google** returned to one of your queries (without taking a glance at the rest papers) ?

Exploratory Search

Wikipedia:

"**Exploratory search** is a specialization of information exploration which represents the **activities carried out by searchers** who are either:

a) unfamiliar with the domain of their goal (i.e. need to learn about the topic in order to understand how to achieve their goal)

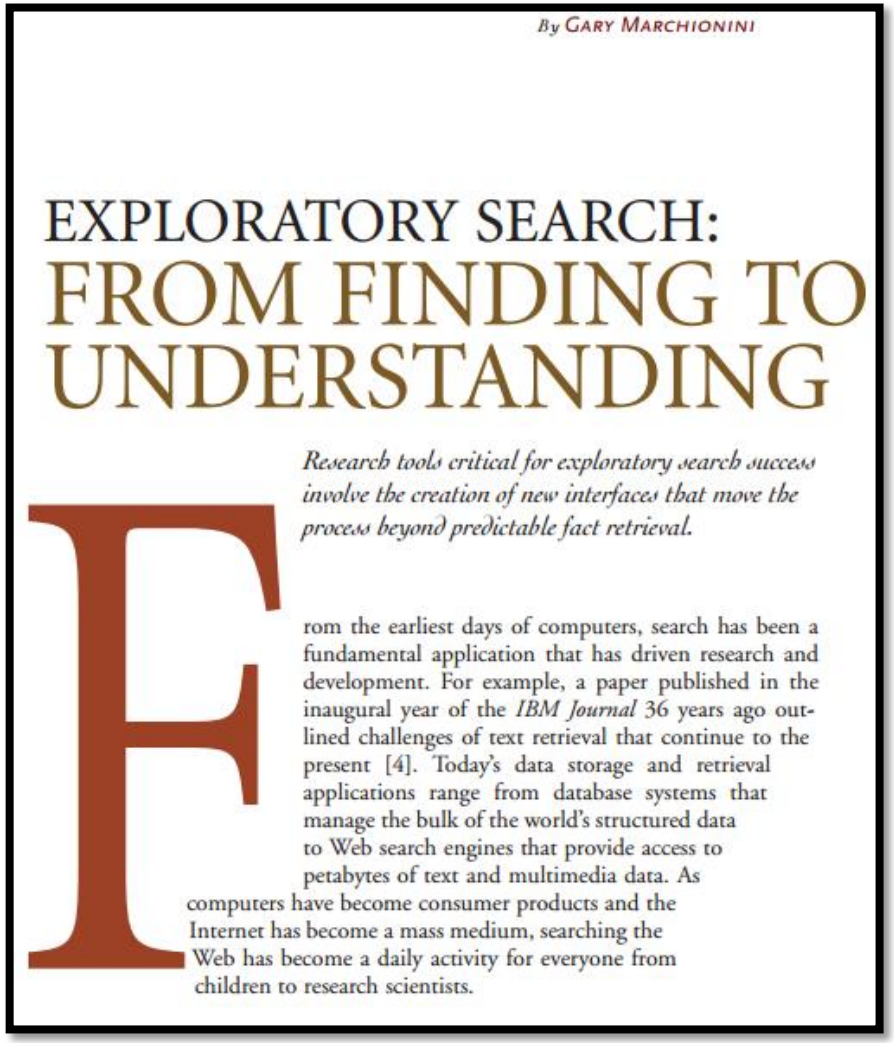
b) unsure about the ways to achieve their goals (either the technology or the process)

c) or even unsure about their goals in the first place.

Consequently, **exploratory search** covers **a broader class of activities** than typical information retrieval, such as ***investigating, evaluating, comparing, and synthesizing***, where new information is sought in a defined conceptual area; exploratory data analysis is another example of an information exploration activity. Typically, therefore, such users generally **combine *querying* and *browsing*** strategies to foster learning and investigation."

Therefore…

- Ranking is not enough for exploratory search



By GARY MARCHIONINI

EXPLORATORY SEARCH: FROM FINDING TO UNDERSTANDING

Research tools critical for exploratory search success involve the creation of new interfaces that move the process beyond predictable fact retrieval.

From the earliest days of computers, search has been a fundamental application that has driven research and development. For example, a paper published in the inaugural year of the *IBM Journal* 36 years ago outlined challenges of text retrieval that continue to the present [4]. Today's data storage and retrieval applications range from database systems that manage the bulk of the world's structured data to Web search engines that provide access to petabytes of text and multimedia data. As computers have become consumer products and the Internet has become a mass medium, searching the Web has become a daily activity for everyone from children to research scientists.

Some Common Requirements for Effective Exploratory Search

- Allow browsing and inspecting the found hits in **groups** (according to various criteria)
- Allow easy and fast access even to **low ranked** hits
- Offer **overviews** of the search results
  - Compute and show descriptions and **count** information for the various groups, or other **aggregated** values
- Allow **gradual** restriction/ranking of the search results

Faceted Search/Exploration

Faceted Exploration is a **widely** used interaction scheme **for Exploratory Search**

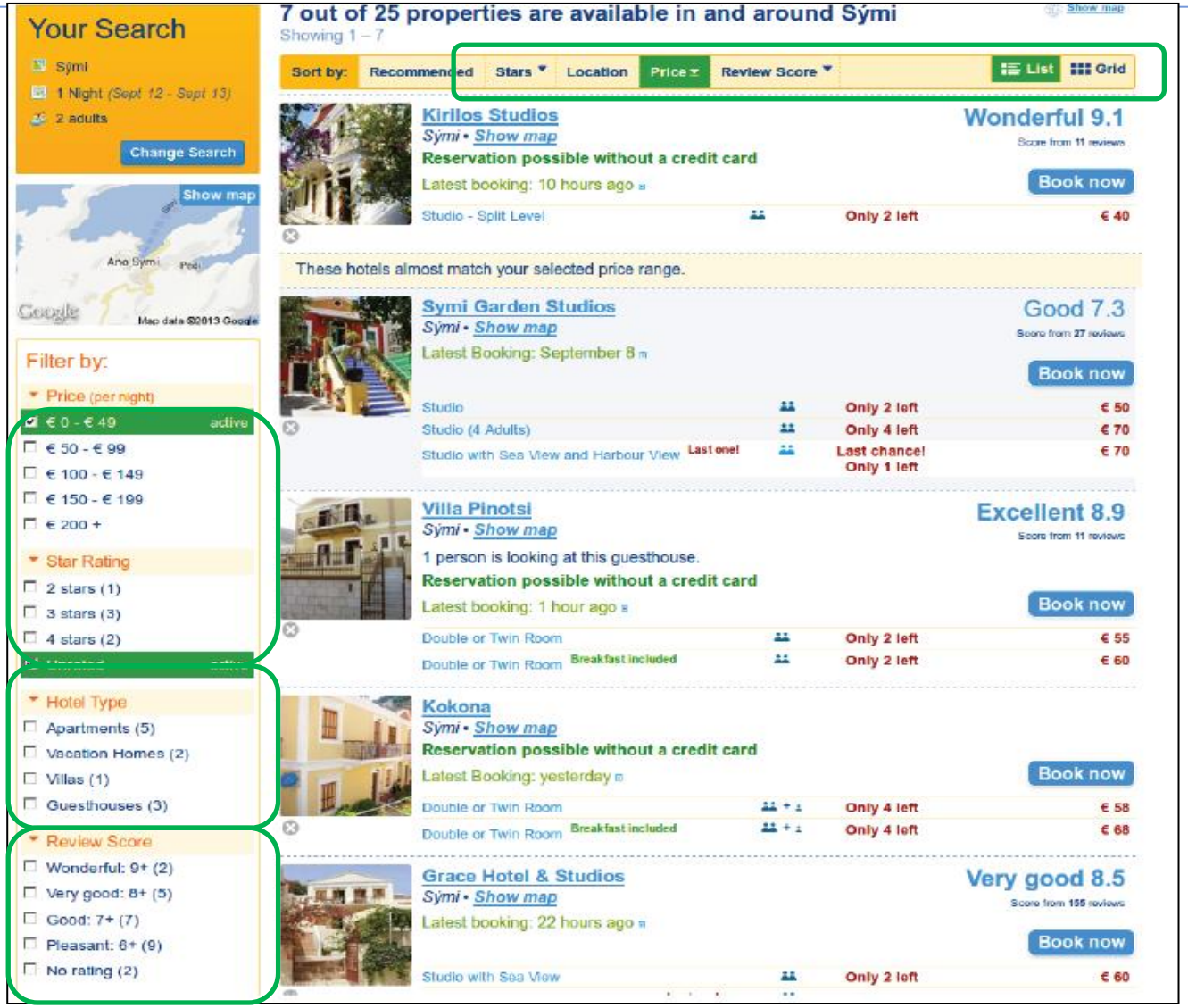
A short (and rather informal) definition:

FE is a **session-based** interactive method for **query formulation** (commonly over a multidimensional information space) through **simple clicks,** that offers

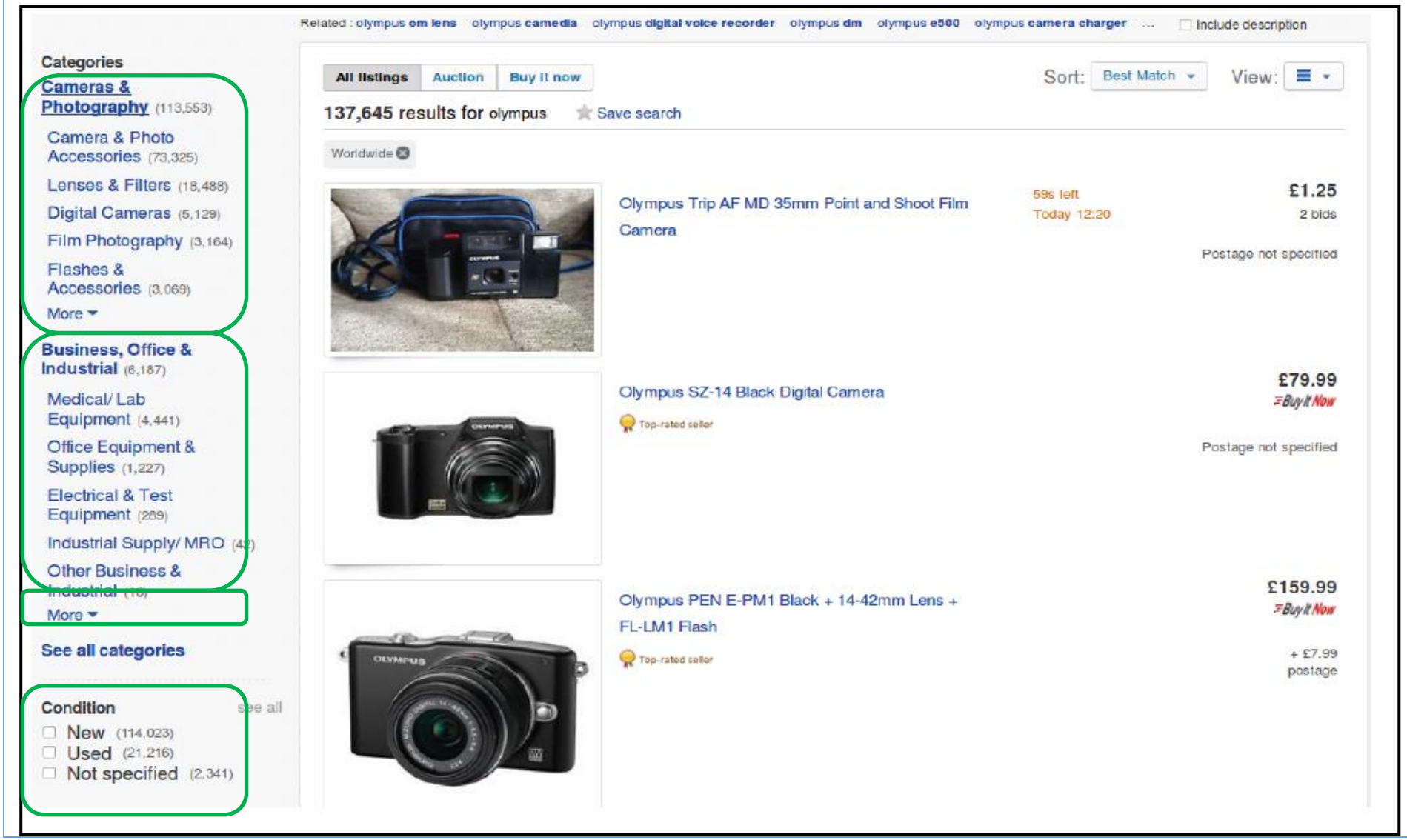
- ✓ an **overview** of the result set (groups and count information)
- ✓ **never leads to empty result** sets

Let's now see some examples from some widely used systems that support faceted search

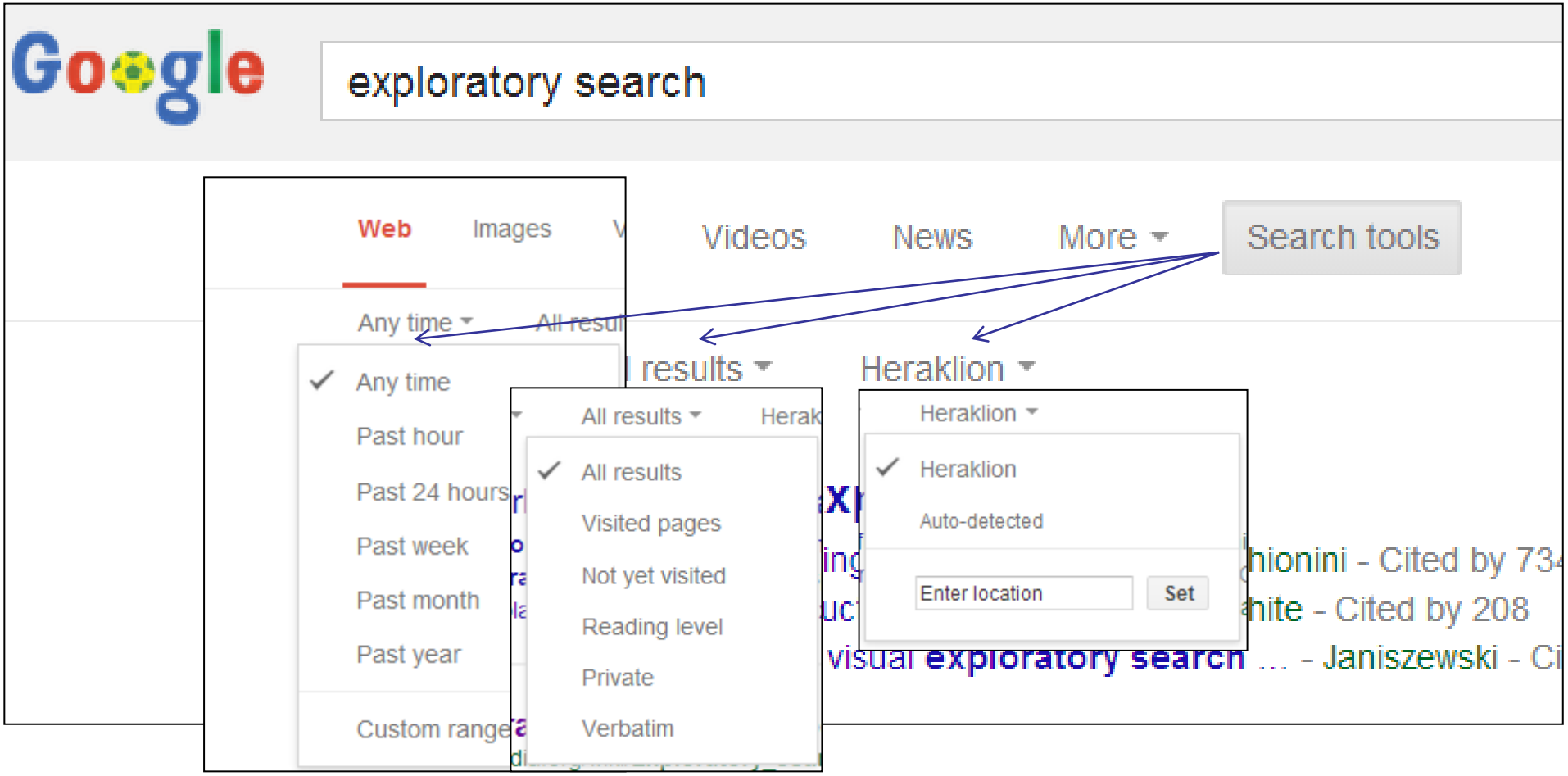
Example of FDT: **Booking**.com



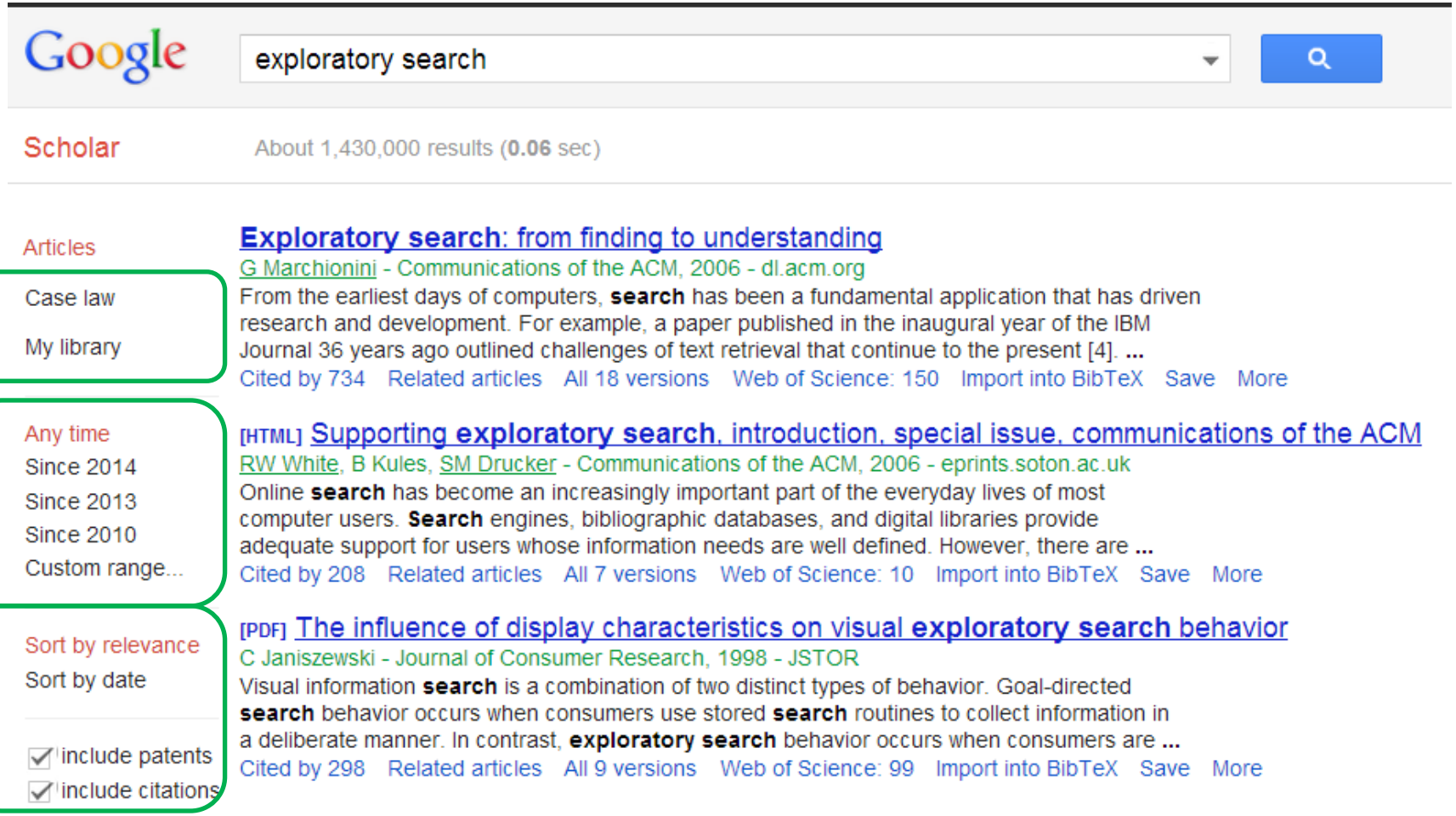
Example: ebay



Example: **Google search** (limited functionality: no count information)



Example: **Scholar Google**



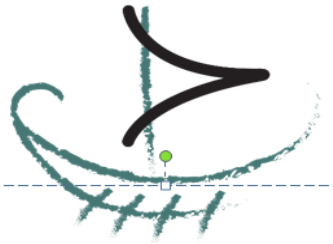
There are plenty of approaches and systems that Support Faceted Search

RB++ [45]
Elastic Lists [46]
Flexplorer [47] and Mitos [23]
Flamenco [31]
Faceted search impl. [14]
Dynamic Faceted Search syst. [15]
Hippalus [2]
IOS/XSearch []
Google Scholar/Seach Tools []

Longwell [48]
mspace [49]
Ontogator [50]
MuseumFinland [51]
Camelis2 [22]
Faceted Data Explorer [52]
NFB [53]
GRQL [41]
/facet [32]
Humboldt [42]
VisiNav [21]
Parallax [54]
Faceted Wikipedia [55]
MediaFaces [56]
BrowseRdf [26]
Fuzzy view based search [57]
Odalisque [58]

System: Hippalus (2013-)

Hippalus: Preference-enriched Faceted Exploration



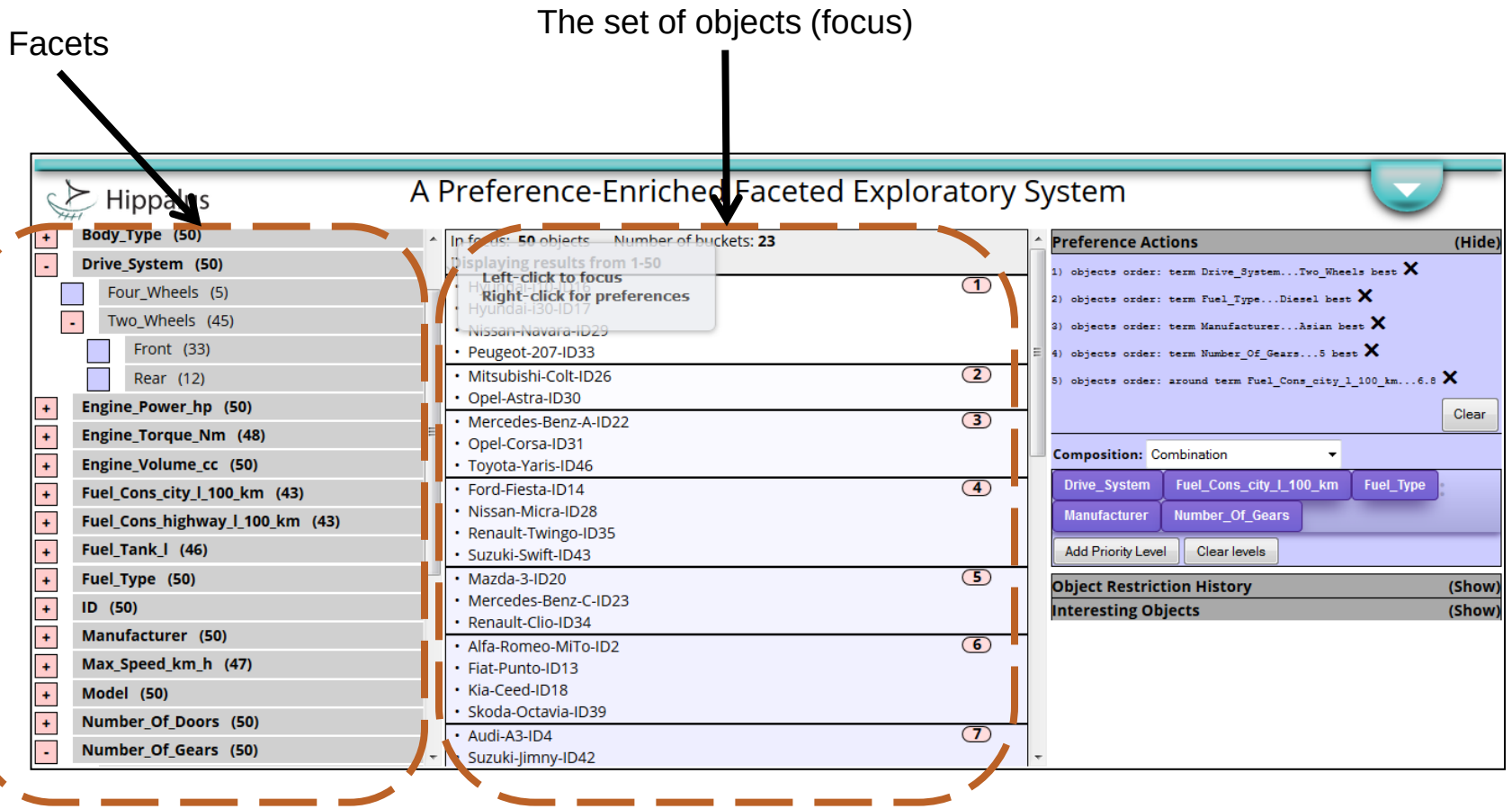
P. Papadakos[1,2] and Y. Tzitzikas[1,2]

First we shall see the faceted search functionality.
(the *preferences* later on)


RUBICON


WHAT NEW CAR SHOULD I BUY?
WHY ARE YOU BUYING IT?
TRANSPORT
FUN
WHAT DO YOU CARE ABOUT?
BEING ACTIVE/DOING THINGS
WHAT PEOPLE THINK OF ME
ALL THE MONEY I DON'T HAVE
THE ENVIRONMENT
MY GROWING FAMILY
FORD FOCUS
HONDA FIT
WHAT DO YOU WANT THEM TO THINK?
WHAT IF PEOPLE KNOW YOU'RE BROKE?
FINE, WHATEVER
FORD FIESTA
CHEVY SPARK
NISSAN VERSA NOTE
I'LL DISAPPEAR
HONDA CIVIC
MERCEDES CLS
AUDI A7
LOVE IT
HOPE YOU LIKE DEBT
I SEEM RICH
NO, I REALLY AM RICH
I'M WEIRD
I'M A GANGSTER OR SOMETHING
CHRYSLER 300
BUT I DON'T
MAZDA6
HYUNDAI GENESIS
KIA OPTIMA
DO YOU LIKE THE SKY?
EH, IT'S OKAY
OH YEAH
WHAT KIND OF WEIRD?
ASYMMETRIC
HYUNDAI VELOSTER
JAGUAR XJ
AUDI A8
GOD, YES
ARE YOU A BRAND SNOB?
BENTLEY CONTINENTAL GT CONVERTIBLE
THEY STILL SELL THOSE?
MITSUBISHI LANCER
NAH, WHO CARES
LINCOLN MKZ
INFINITI M37
OLD SCHOOL
VW BEETLE
FIAT 500
KAWAII
SCION iQ
NISSAN CUBE
RUN ON WHAT?
SOME ELECTRICITY
DIESEL
THAT STUFF THAT WORKED SO WELL IN THE HINDENBERG
BATTERIES
CHEVY VOLT
VW JETTA SPORTWAGON TDI
HONDA FCX CLARITY
NISSAN LEAF
FORD FUSION HYBRID
HOW ABOUT A KOREAN TAKE?
SURE, WHY NOT
KIA SOUL
CAN'T A PARENT
OPEN THE HATCH BY KICKING THE AIR
FORD C-MAX
HAVE A COOL CAR
BMW X1
MORGAN 3-WHEELER
BEING AWESOME
GOING FAST
CARVING CURVES
GOING SLOW, FEELING FAST
WHAT'S FUN FOR YOU?
HOW MANY CYLINDERS?
WHERE DO YOU WANT YOUR ENGINE?
6
8
12
WANT A STICK?
YEP
ASTON MARTIN VANTAGE
BEHIND ME
IN FRONT OF ME
PORSCHE 911
NISSAN GT-R
HOW MANY PEOPLE ARE WITH YOU?
IS 12 TOO MANY, BUT 8 TOO FEW?
YEP
LAMBORGHINI GALLARDO
SRT VIPER
AUDI R8
I FORGOT, I NEED A USABLEBACK SEAT
0-1
2-3
4+
CADILLAC CTS-V WAGON
WANT TO GO FAST WITH 4+ PEOPLE WITHOUT GAS?
SURE
TESLA MODEL S
BMW 335i
CADILLAC ATS
ARE YOU LOADED?
WANT TO PEEL OUT AT A DAIRY QUEEN?
FUCK YEAH
FORD MUSTANG
CHEVY CAMARO
DODGE CHALLENGER
WHY, YE$
ONLY IF YOU MEAN DRUNK
UM, I'M GOOD
SHOUTY
CLASSY OR SHOUTY?
FERRARI 458
McLAREN MP4-12C
CLASSY
JAGUAR F-TYPE
CHEVROLET CORVETTE STINGRAY
AXIS OR ALLIES?
AXIS
ALLIES
BOTH?
BMW M5
MERCEDES E63 AMG
PORSCHE PANAMERA
CADILLAC CTS-V
MASERATI GHIBLI
SUBARU WRX
PORSCHE BOXSTER
LOTUS ELISE
PORSCHE CAYMAN
NONE OF YOUR BUSINESS
NO!
HAVE A HAIRPIECE?
NAH
I'M COMFORTABLE
EVER DRIVE ON UNPAVED ROADS?
HOW'S YOUR CASH SITUATION?
SURE DO
AUDI ALLROAD QUATTRO
NOTSO HOTSO
SCION FR-S/SUBARU BRZ
MAZDA MIATA
CATERHAM CLASSIC
FIAT 500 ABARTH
BRING THE NOISE
HOW MUCH DO YOU LIKE NOISE?
MINI COOPER
FORD FIESTA ST
PLUCKY
WHAT "P" EXCITES YOU?
SHH, KEEP IT DOWN
PRECISE
PRACTICAL
PUNCHY
VW GOLF GTI
FORD FOCUS ST
MAZDASPEED3
VOLVO S60
JALOPNIK
CHART BY JASON TORCHINSKY
© 2013

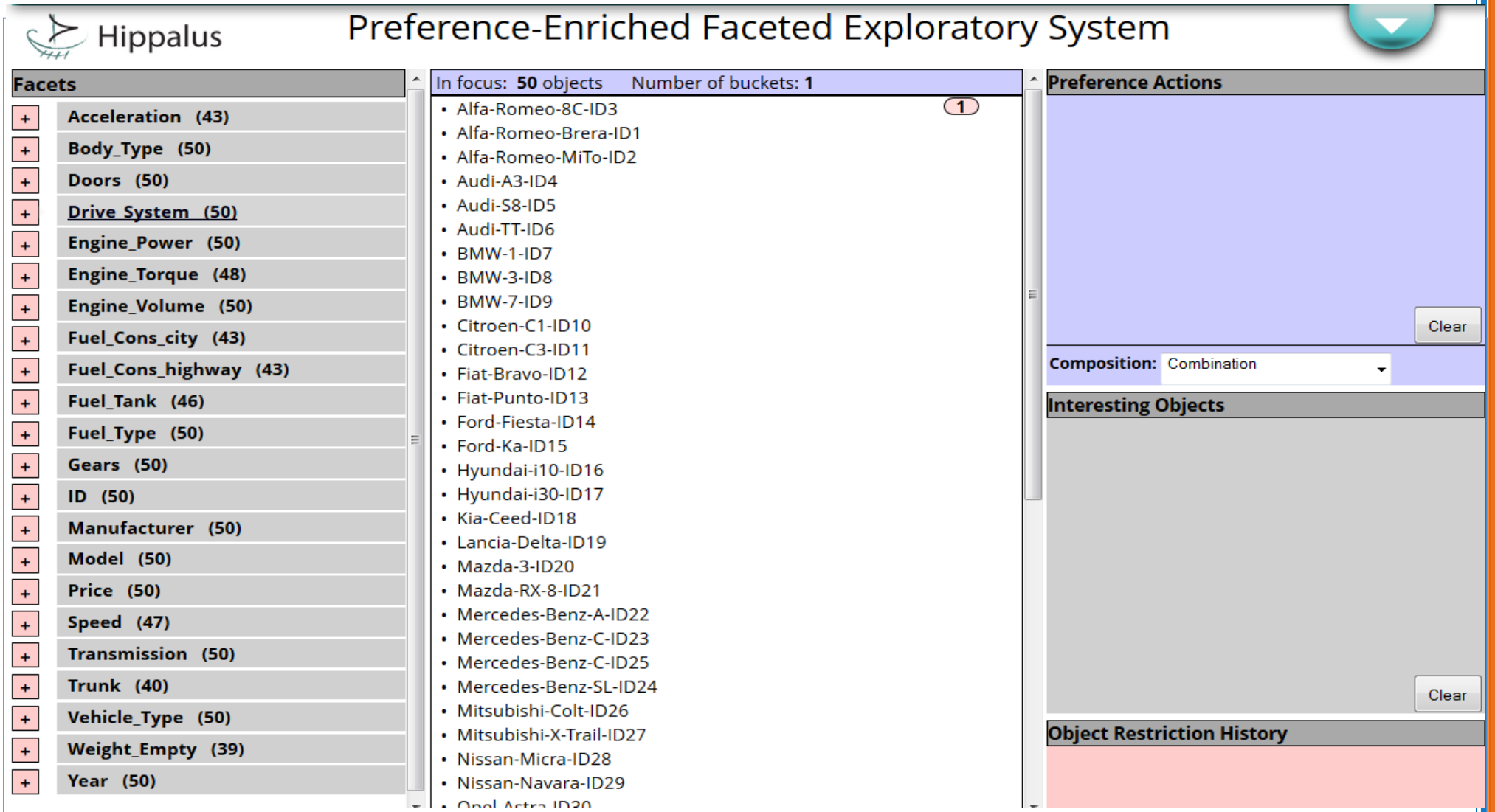
Hippalus: A system offering Faceted Search



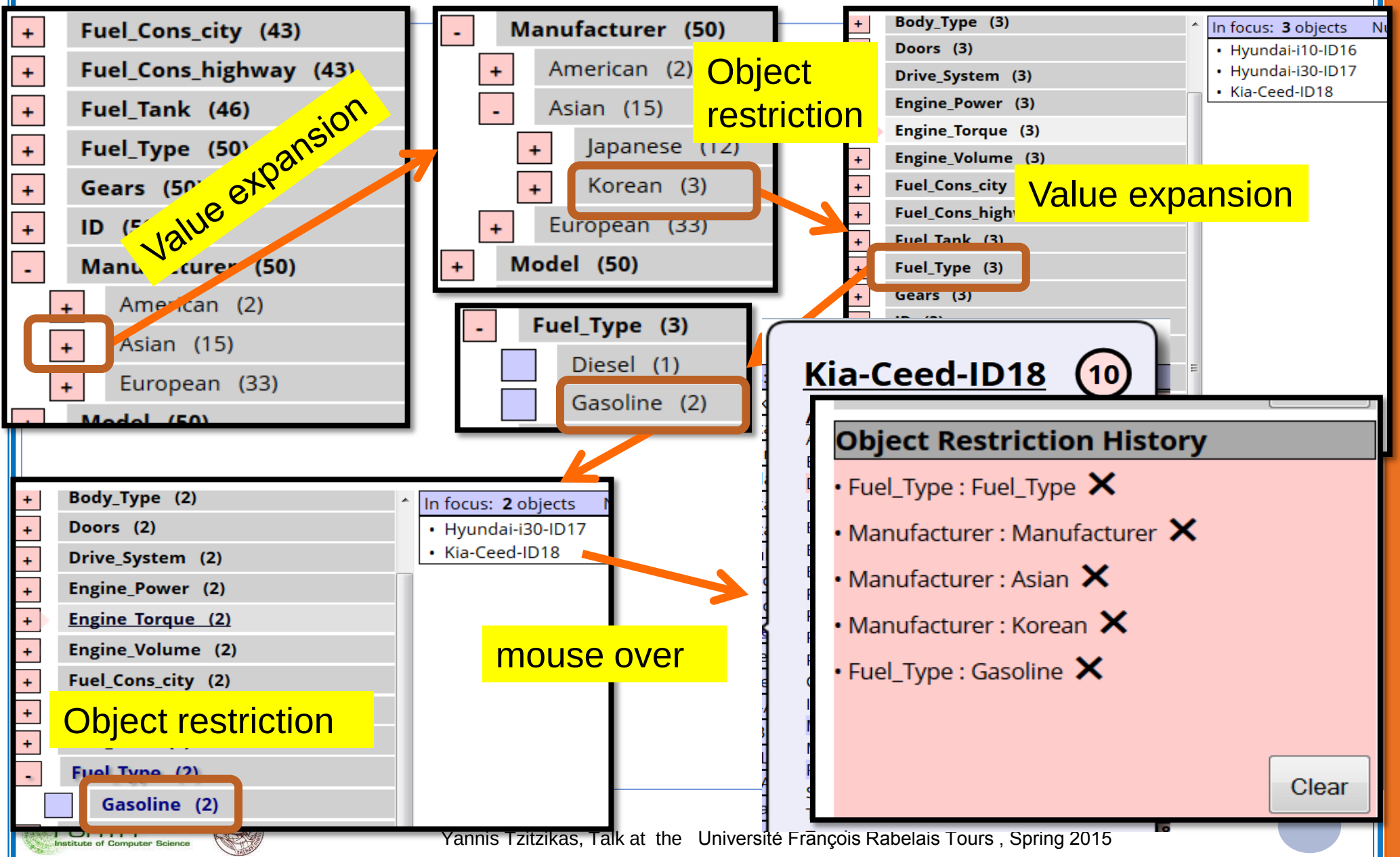
Facet-terms

- Left-click on a term: action that changes the focus

Hippalus: Interaction over a KB of 50 cars



Hippalus: FDT interactions



Preferences

Preferences

Preferences

- are **not** hard constraints
- are **not** necessarily numbers
- are **not** necessarily total orders
- are personalized ***wishes*** "*I like A better than B*"
- may be complex covering multiple attributes

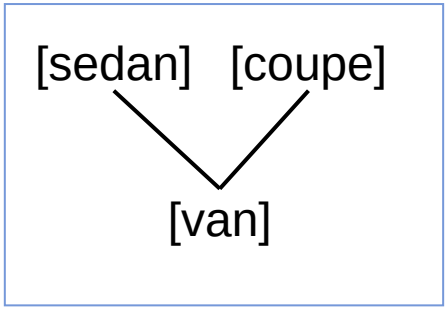
Approaches for **Defining** Preferences (not only in databases)

Two main approaches for specifying preferences

- **Qualitative** Approach
  - In the qualitative approach, the preferences between tuples in the answer of a query are specified directly, typically using a **preference binary relation $>_{pref}$**
  - E.g. a tuple **v** *is more preferred than* a tuple **v'** if **$v >_{Pref} v'$**
- **Quantitative** Approach
  - In the quantitative approach, preferences are specified indirectly using **scoring** functions that associate a numeric score with every tuple of the query answer:
  - E.g. a tuple **v** *is more preferred than* a tuple **v'** if Score(**v**) > Score(**v'**)

Preferences: **Qualitative** vs **Quantitative**

I prefer sedan to van
I prefer coupe to van



≠

Score(sedan)=2
Score(coupe)=2
Score(van) =1

- The **qualitative** approach is more powerful (in terms of expressive power) than the **quantitative** one
- Moreover, there is no obvious method users could follow for specifying and combining scores.

In brief, the qualitative method is more expressive and convenient for the user, however the evaluation of preference-aware query answers is sometimes more expensive.

Preferences and Databases

- Thoroughly studied in the **Database world** but usually
  - Users must be **acquainted** with the **information space** and **available choices** for expressing their preferences
  - **Hierarchically organized attribute values** and **multi-valued** attributes are not supported
  - Preferences are given in **one shot** (not **gradually**)
  - Sometimes users have to formulate **complicated queries** or use **complex UIs**

Our objective (as we shall see):

- Tackle all the above shortcomings by extending Faceted Exploration with preferences

Preference-enriched Faceted Search

Outline

- Introduction and Background
  - Exploratory Search
  - Faceted Exploration
- Preferences
- Preference-enriched Faceted Search (PFS)
  - Introduction
  - Demo Hippalus
  - Formally
    - FS
    - PFS
    - Algs
  - Evaluation
  - Current Investigations
- Concluding Remarks
- References
- Acknowledgements

Objectives and Requirements

- Objectives
  - To provide a **theoretical framework** for preferences over multi-dimensional and hierarchical information spaces (including set-valued descriptions)
  - To extend the **interaction model** of faceted search with preferences
- Other (Non Functional) Requirements
  - The extension should be easy to use
    - Without requiring from the user to type anything, without having to be familiar with the information space
  - The extension should be applicable to large information bases
    - The Multidimensional information spaces with hierarchies and set-valued data can capture various cases: a relational table, the results of a SPARQL Query, an information space derived by a process from text mining,etc

PFS: Preference-enriched Faceted Search

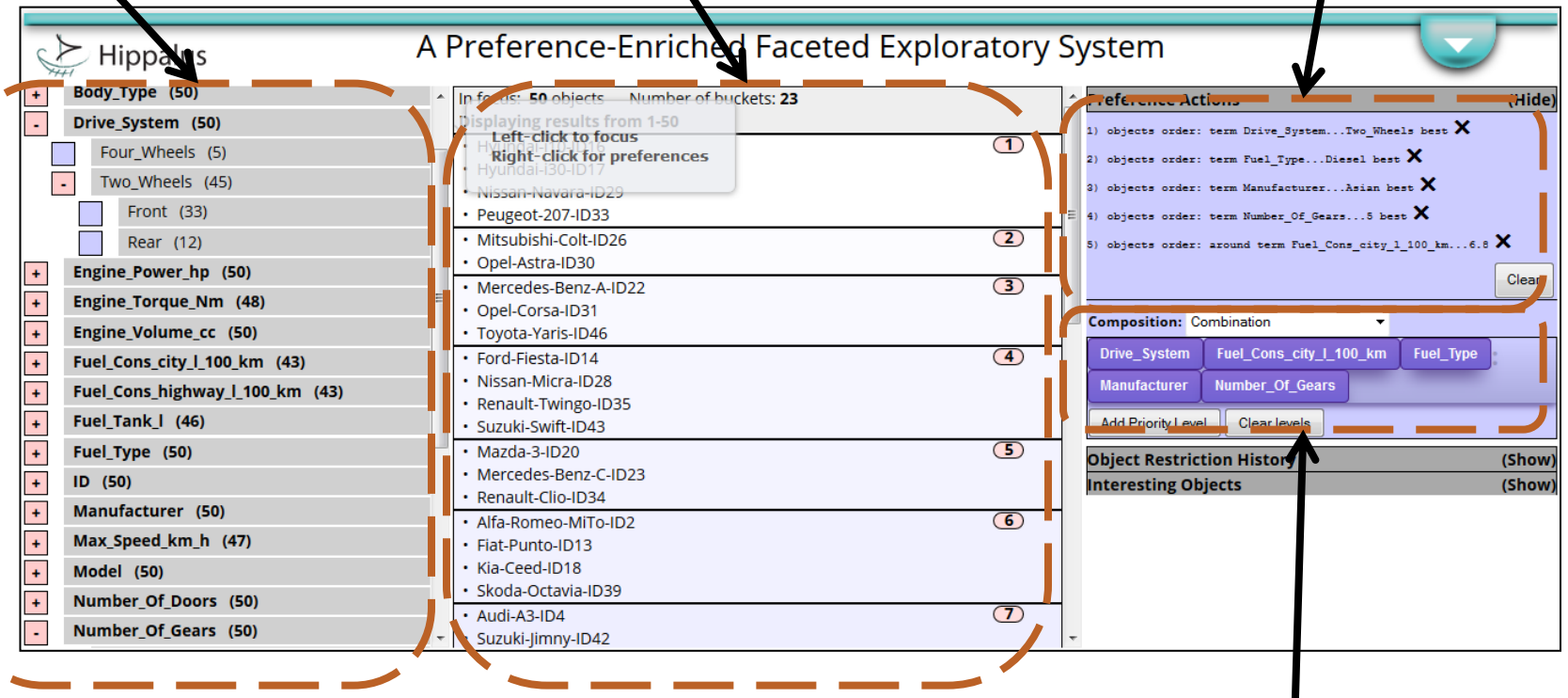
- An extension of the interaction paradigm of Faceted Exploration with actions that allow the users to express at browsing time their preferences.
  - The user has to kinds of actions
    - Actions that change the focus (zoom-in/out/side actions), as in classical Faceted Search
    - Actions that rank the focus (based on preferences)
- The proposed model supports progressive preference expression, inherited preferences and automatic scope-based resolution of conflicts over single or multi-valued attributes with hierarchically organized values.
- Algorithms enabling the application of the model over large information bases.

Hippalus and Preferences

Facets

The set of objects (focus) ranked according to the **expressed preferences**

History of preference actions

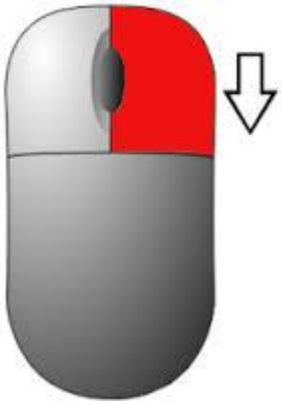


Facet-terms

- Left-click: action that changes the focus
- **Right-click: actions for expressing preferences**

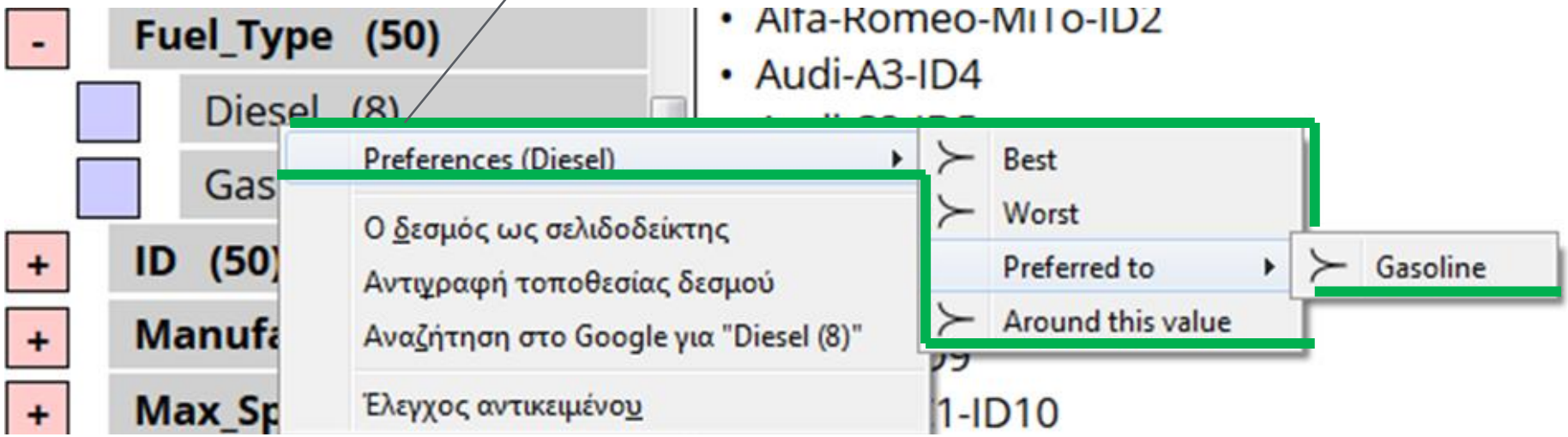
Expression of preference priorities

Right-click on the value `Diesel` of the facet `Fuel_Type`



`Diesel > Gasoline`

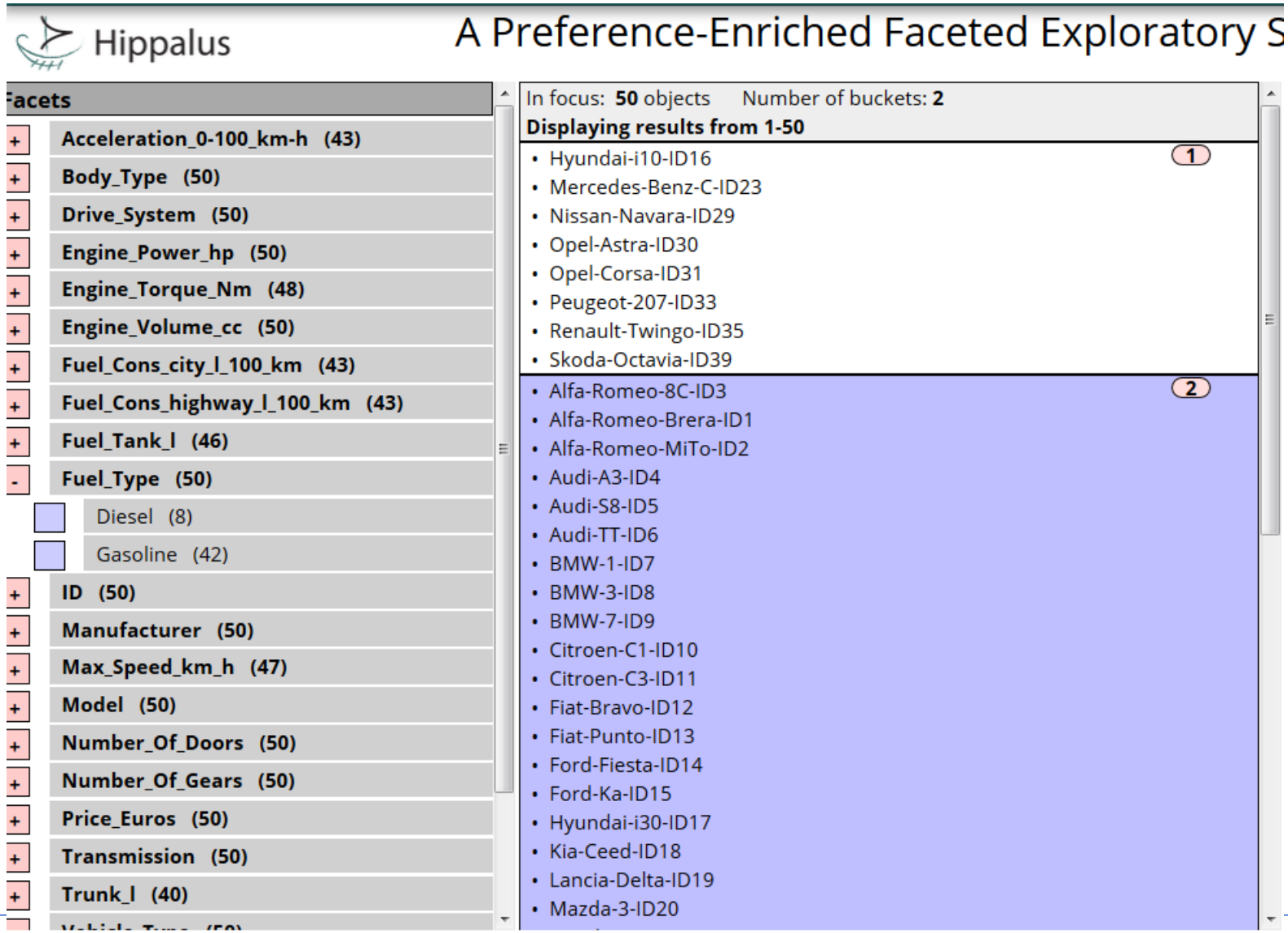
(meaning that I prefer Diesel to Gasoline)



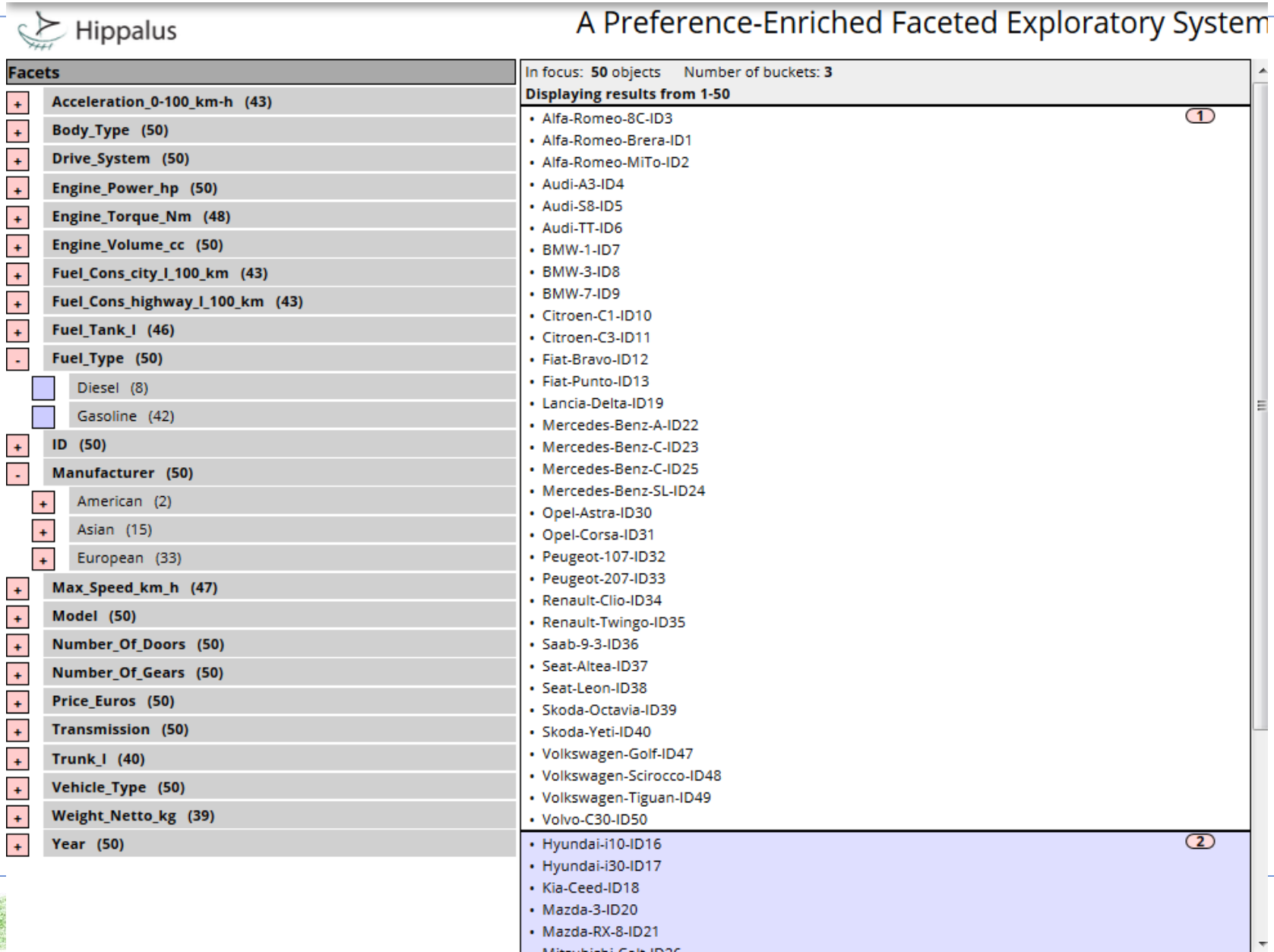
Live Demo

- Script A
  - Prefer Diesel to Gasoline
    - For making clear the difference between faceted search and PFS
- Clear Prefs
  - Prefer European to Asian
    - For making clear the benefit of hierarchies and inherited preferences

Diesel > Gasoline



European > Asian



How it works

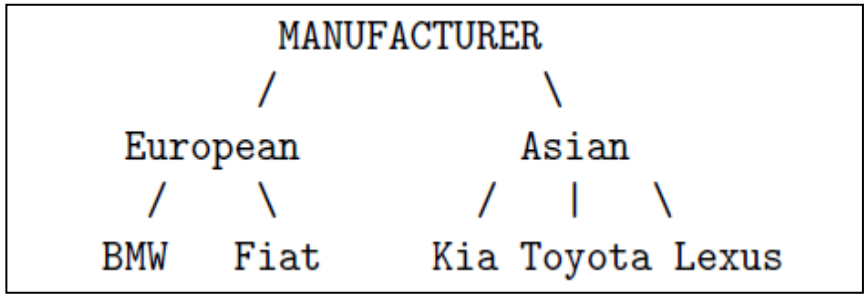
Background

- A binary relation over a set A is any subset R of AxA
- A binary relation is a partial order it is reflexive, transitive and antisymmetric.
- A binary relation is a total order if it is a partial order and total
  - A relation is total if for every a,b in A, either aRb or bRa
- A linear extension of a binary relation R is a total order L such that $R \subseteq L$.
  - A total order T is a linear extension of a partial order R if, whenever aRb it also holds that aLb
- A bucket order is a linear order of subsets.
  - Example of a bucket order with four blocks:
    - **<{a,b,c}, {d}, {e,f}, {g,h}>**

Taxonomy

- A taxonomy Ti is a pair (Ti,≤i), where Ti is a set of terms and ≤i a partial order over Ti.

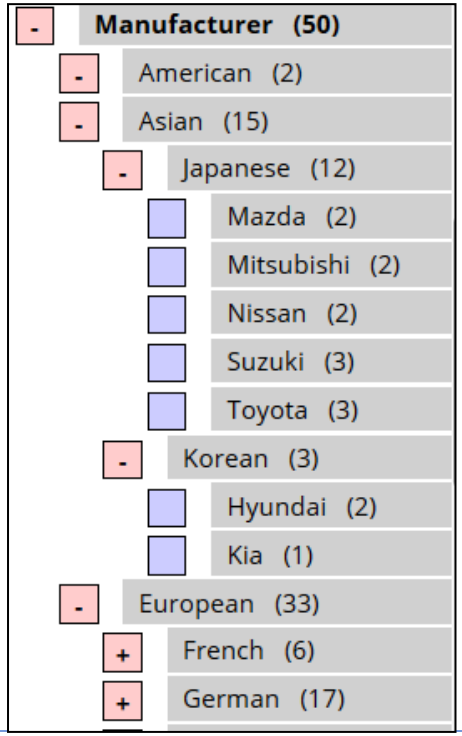
Example



T= { MANUFACTURER, European, Asian, BMW, Fiat, Kia, Toyota, Lexus}
≤ = {

BMW ≤ European,
Fiat ≤ European,
European ≤ MANUFACTURER,
Asian ≤ MANUFACTURER,
Kia ≤ Asian,
Toyota ≤ Asian,
Lexus ≤ Asian,

}
The reflexive and transitive relationships have been omitted

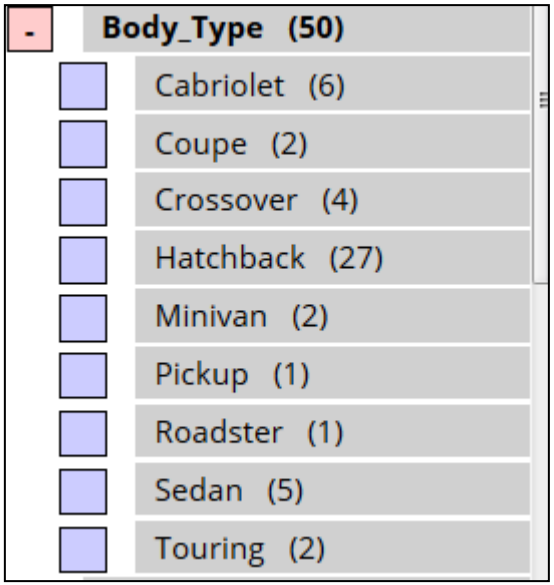


Taxonomy (cont'd)

- If $\leq = \varnothing$ then the taxonomy is flat

Example

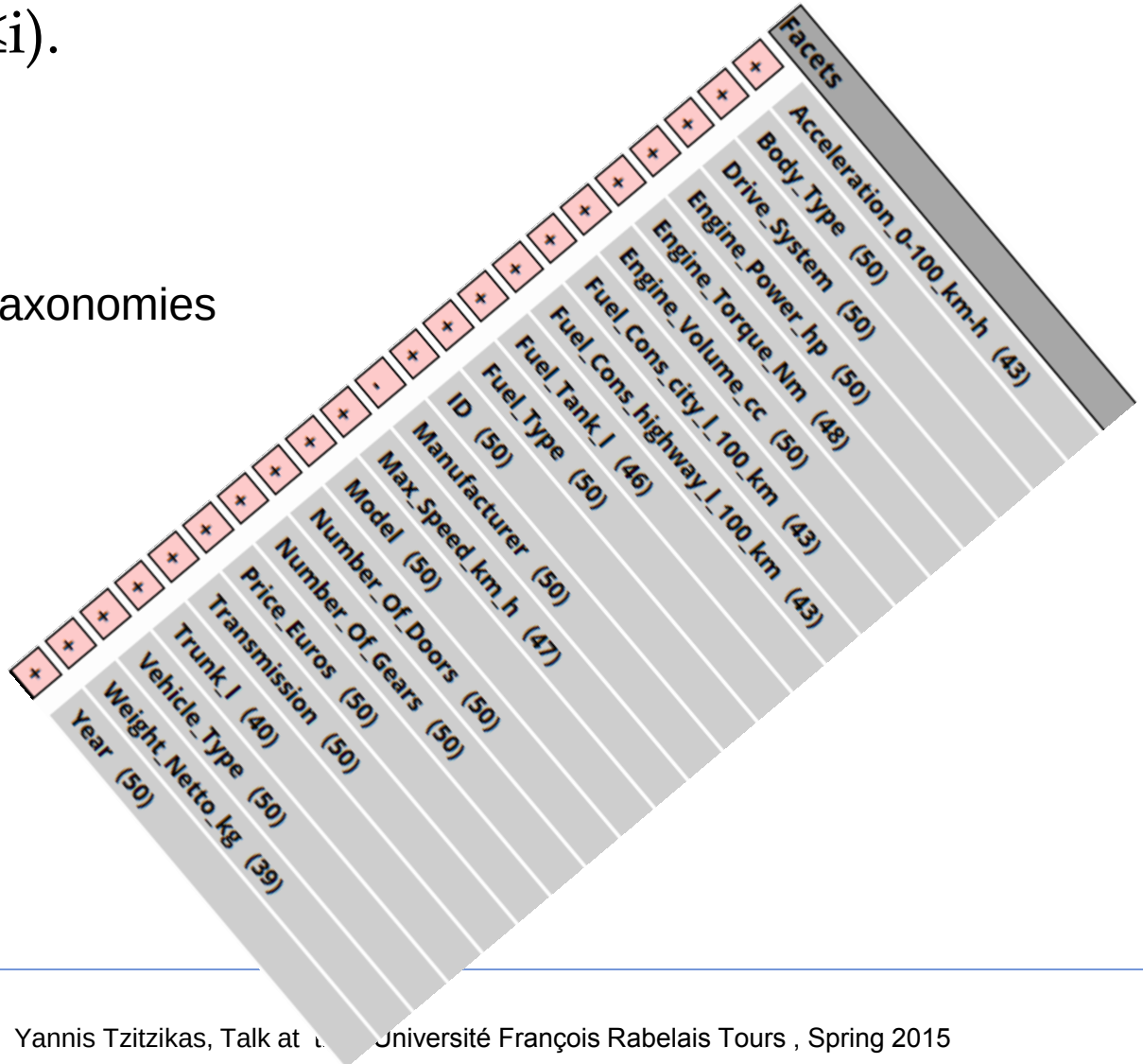
T = { Cabriolet, Coupe, Crossover, Hatchback, Minivan, Pickup, Roadster, Sedan, Touring}
$\leq = \varnothing$



Faceted Taxonomy

- A faceted taxonomy is a set of taxonomies, i.e. F={F1, …, Fk} where Fi=(Ti,≤i).

Example
In our demo we have 22 taxonomies



Descriptions of Objects

- Let Obj be the set of objects of interest (e.g. the cars in our demo)
- Each object o ∈ Obj is described (wrt a faceted taxonomy F={F1, …, Fk} where Fi=(Ti,≤i)) by associating it (classifying it) with one or more (or none) terms from each Ti.
- Let **o** denote the description of o

Example

Assume F = {Manufactuer, BodyType},

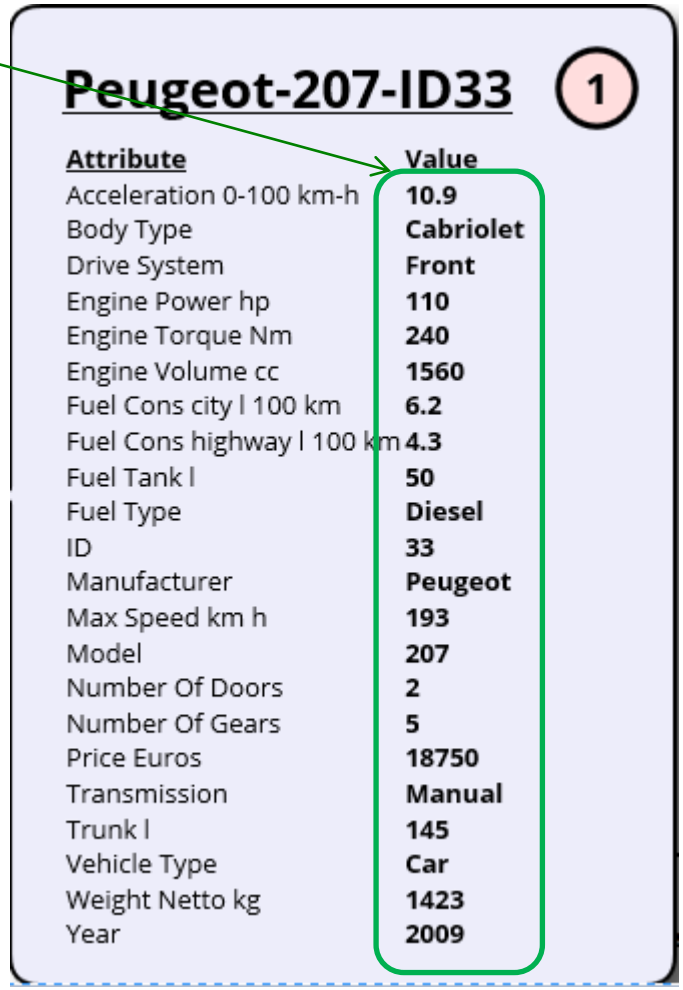
and that Obj={o1, o2}. Their descriptions could be

o1 = {Fiat, Minivan}

o2 = {Lexus, Sedan}

Descriptions of Objects (cont'd)

- Peugeut-207-ID33 ∈ Obj
- **Peugeut-207-ID33 =**



Peugeot-207-ID33 (1)

Attribute	Value
Acceleration 0-100 km-h	**10.9**
Body Type	**Cabriolet**
Drive System	**Front**
Engine Power hp	**110**
Engine Torque Nm	**240**
Engine Volume cc	**1560**
Fuel Cons city l 100 km	**6.2**
Fuel Cons highway l 100 km	**4.3**
Fuel Tank l	**50**
Fuel Type	**Diesel**
ID	**33**
Manufacturer	**Peugeot**
Max Speed km h	**193**
Model	**207**
Number Of Doors	**2**
Number Of Gears	**5**
Price Euros	**18750**
Transmission	**Manual**
Trunk l	**145**
Vehicle Type	**Car**
Weight Netto kg	**1423**
Year	**2009**

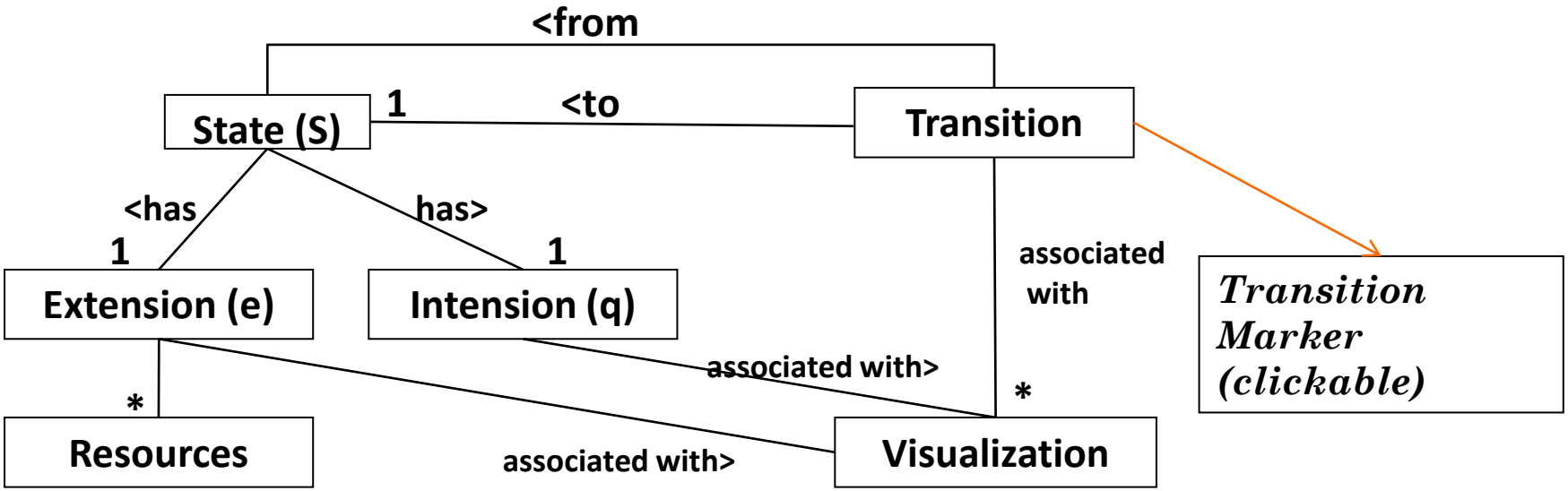
Interpretation of Facets' Terms

- I(Peugeut) = {Pegeut-207-ID33}
- I(European) = ∅

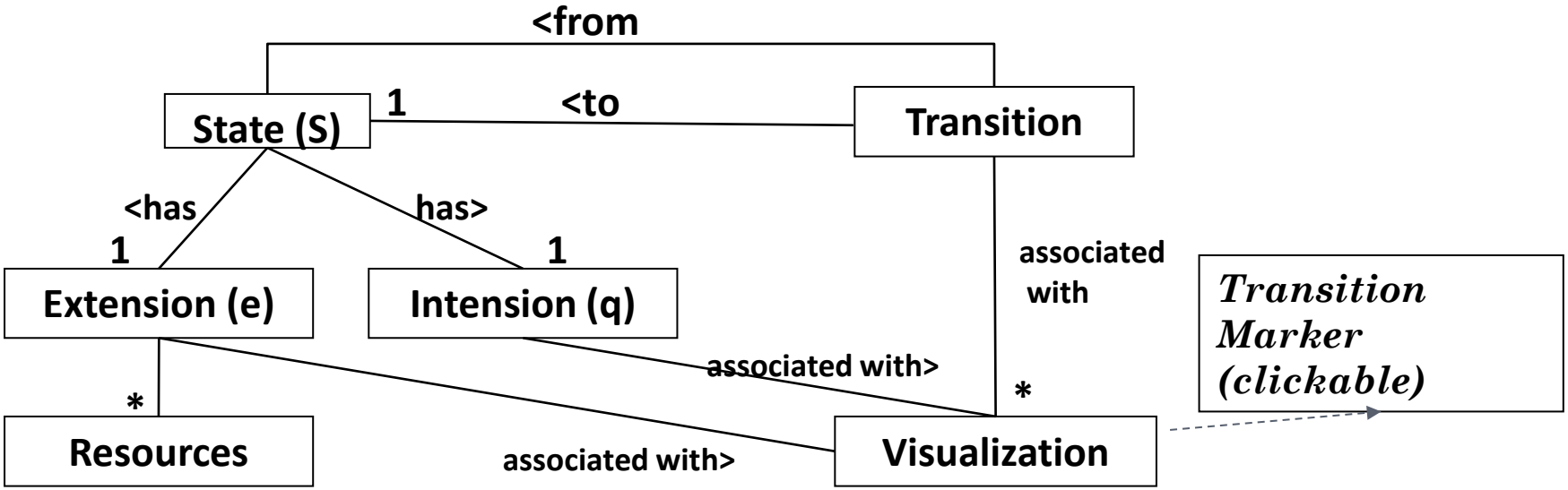
By considering the semantics of ≤ we can define the **model interpretation I**

- **I**(t) = ∪ { I(t') | t' ≤ t}
- E.g. if Peugeut ≤ European then
- **I**(Peugeut) = I(Peugeut) = {Pegeut-207-ID33}
- **I**(European) = I(European) ∪ I(Peugeut) ={Pegeut-207-ID33}

Faceted Search Interaction (abstract)



Faceted Search Interaction (cont'd)



- Only terms that if added to the current query (intension) of the current state will yield to no empty results are shown (as Transition Markers).
- On click the user goes to the new state

Formal Synopsis: Taxonomy

TAXONOMY		
Name	**Notation**	**Definition**
terminology	T	a set of *terms* (can capture categorical/numeric values)
subsumption	$\leq$	a partial order (reflexive, transitive and antisymmetric)
taxonomy	$(T, \leq)$	T is a terminology, $\leq$ a subsumption relation over T
broaders of t	$B^+(t)$	$\{ t' \mid t < t' \}$
narrowers of t	$N^+(t)$	$\{ t' \mid t' < t \}$
direct broaders of t	$B(t)$	$minimal_<(B^+(t))$
direct narr. of t	$N(t)$	$maximal_<(N^+(t))$
Top element	$\top_i$	$\top_i = maximal_\leq(T_i)$

Formal Synopsis: Materialized Faceted Taxonomy

MATERIALIZED FACETED TAXONOMIES		
faceted taxonomy	$\mathcal{F}=\{F_1,...,F_k\}$	$F_i=(T_i,\leq_i)$, for $i=1,...,k$ and all T_i are disjoint
object domain	Obj	any denumerable set of objects
interpretation of T	I	any function $I: T\rightarrow 2^{Obj}$
materialized faceted taxonomy	$(\mathcal{F},I)$	$\mathcal{F}$ is a faceted taxonomy $\{F_1,...,F_k\}$ and I is an interpretation of $T=\bigcup_{i=1,k} T_i$
ordering over interpretations	$I\sqsubseteq I'$	$I(t)\subseteq I'(t)$ for each $t\in T$
model of $(T,\leq)$ induced by I	$\bar{I}$	$\bar{I}(t)=\cup\{I(t') \mid t'\leq t\}$
Descr. of o wrt I	$D_I(o)$	$D_I(o)=\{\, t\in T \mid o\in I(t)\}$
Descr. of o wrt $\bar{I}$	$D_I(o)\equiv \bar{D}_I(o)$	$\{\, t\in T \mid o\in \bar{I}(t)\}=\cup_{t\in D_I(o)}(\{t\}\cup B^+(t))$

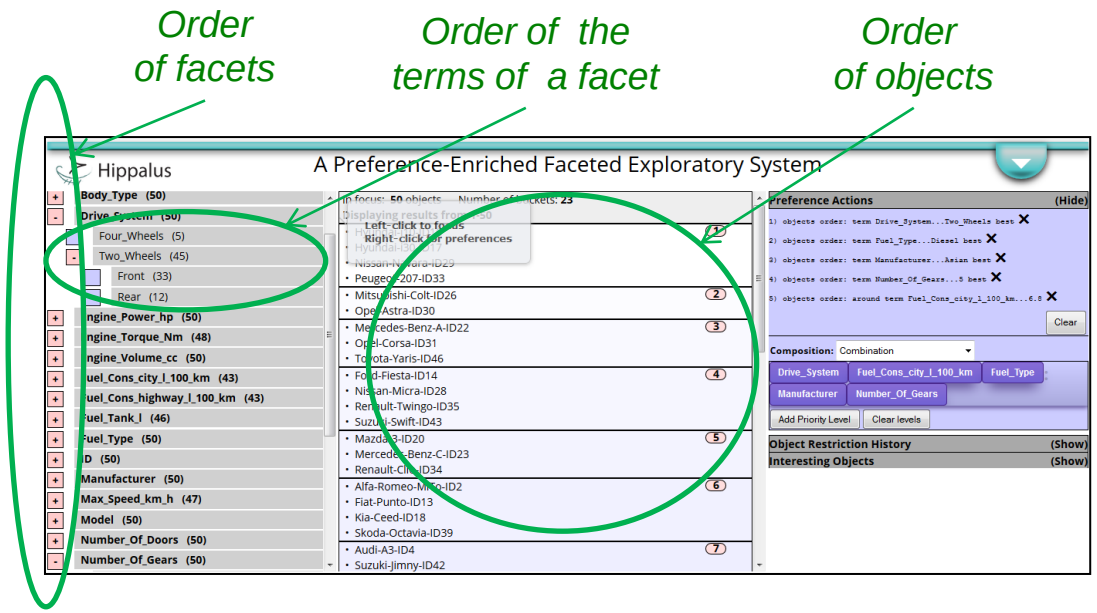
Formal Synopsis: Faceted Search Interaction

FDT-INTERACTION: BASIC NOTIONS AND NOTATIONS		
focus	ctx	any subset of T such that $ctx = minimal(ctx)$
projection on F_i	ctx_i	$ctx \cap T_i$
Kinds of zoom points w.r.t. a facet i *while being at* ctx		
zoom points	$AZ_i(ctx)$	$\{ t \in T_i \mid \bar{I}(ctx) \cap \bar{I}(t) \neq \emptyset \}$
zoom-in points	$Z_i^+(ctx)$	$AZ_i(ctx) \cap N^+(ctx_i)$
immediate zoom-in points	$Z_i(ctx)$	$maximal(Z_i^+(ctx)) = AZ_i(ctx) \cap N(ctx_i)$
zoom-side points	$ZR_i^+(ctx)$	$AZ_i(ctx) \setminus \{ctx_i \cup N^+(ctx_i) \cup B^+(ctx_i)\}$
immed. zoom-side points	$ZR_i(ctx)$	$maximal(ZR^+(ctx))$
Restriction over an object set $A \subseteq Obj$		
reduced interpretation	I_A	$I_A(t) = I(t) \cap A$
reduced terminology	T_A	$\{ t \in T \mid \bar{I}_A(t) \neq \emptyset \} =$ $\{ t \in T \mid \bar{I}(t) \cap A \neq \emptyset \} = \cup_{o \in A} B^+(D_I(o))$

Enriching Faceted Search with Preferences

- During the interaction the user can make **actions,** i.e. a set B={b0, … , bN} that express preferences. These actions determine the ordering of the elements shown These actions are accumulated, i.e. the entire set B affects the ordering.
- These actions determine the ordering of:

facets, facet terms, objects



(Notations)

$>$ $\neq$ $\leq$

Preference

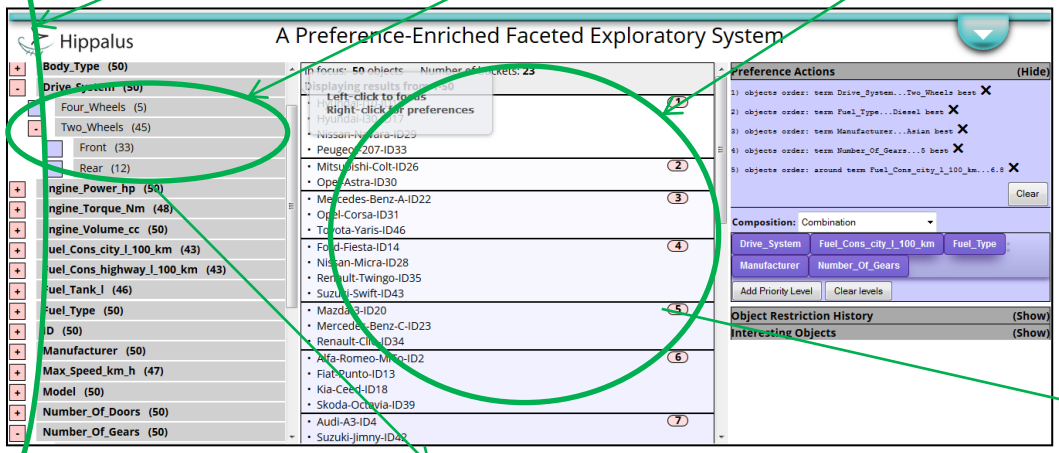
White > Black

BMW > KIA

Subsumption

Paris $\leq$ France

Cont'd



- The more important preference relation (in the sense that the user 's goal is to find objects) is (Obj, >). Hereafter we will focus on that

(F, >)

For each taxonomy (Ti ≤) the user through actions can define a preference relation >i over its terms. Initially >i =∅

(Obj, >)

This can lead to k preference relations, one for each Ti.

(T1, >1) ….. (Tk, >k)

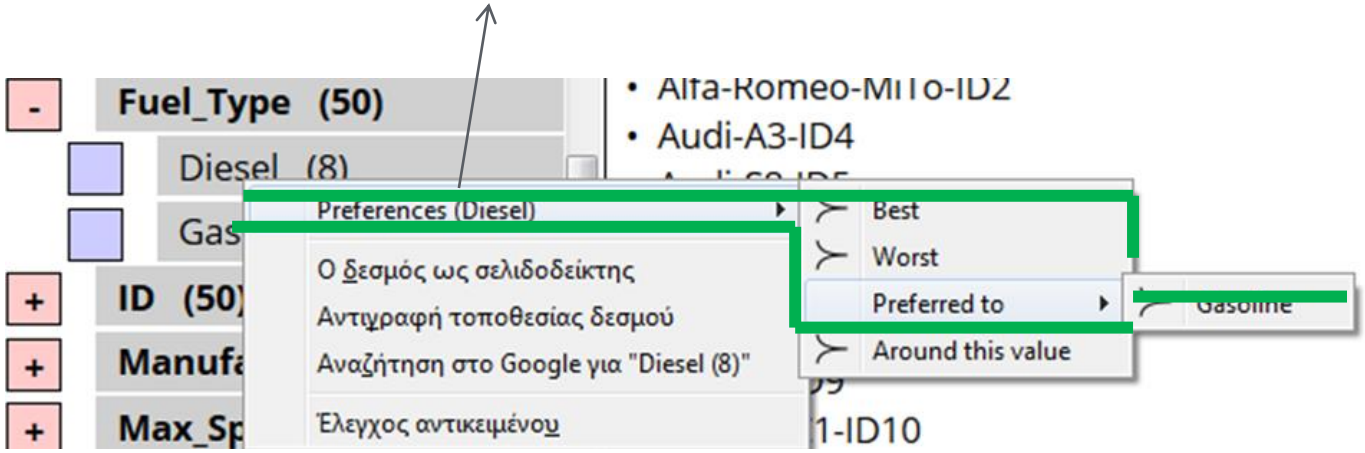
In total k+2 preference relations

Enactment of Preference Actions

- The preference actions are enacted through right clicking on an element
  - this element is called the *anchor* of the action
- Right click actives a pop-up menu and the user selects the preference action that he wants to

`Diesel > Gasoline`

(meaning that I prefer Diesel to Gasoline)



PFS: Theoretical Framework

Relative Preferences actions (over Flat Attributes)

- Consider a facet with a (flat) set of terms **T ={White, Black, Red, Blue}**
- At the beginning there is not preference relation over T, i.e. >=∅
- Suppose the user expresses **a relative preference**, Blue > Red.

- The linear extension of > can be
  - <Blue, {Red,White,Black} > // the inactive elements as minimal
  - <{Blue,White,Black}, Red}> // inactive elements as maximal
  - <Blue, Red, {White,Black} > // inactive elements at the last block
  - It is a matter of policy what of the previous to adopt
    - A system should support more than one policy

Best/Worst actions (over Flat Facets)

Let **T ={White,Black, Red, Blue}**

Example 1:

- Suppose the user expresses the preference **Best**(Blue).
- This is a **shortcut**, i.e. the defined preference relation is
  $> = \{ (Blue > x) \mid x \in T \setminus \{Blue\}\}$
- The linear extension of > is
  - <Blue, {Red, White, Black} >

Example 2:

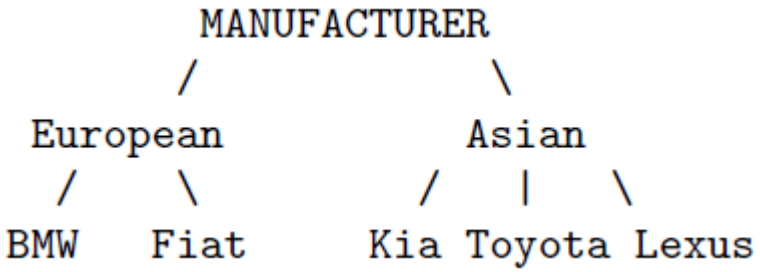
- Suppose the user expresses the preference, **Worst**(Blue).
- This is a **shortcut**, i.e. the defined preference relation is
  $> = \{ (x > Blue) \mid x \in T \setminus \{Blue\}\}$
- The linear extension of > is
  - < {Red, White,Black}, Blue>

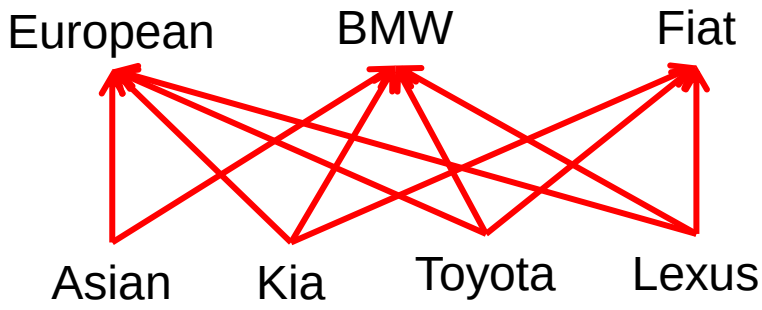
Example 3:

- Best and Worst: Leads to a linear order where the fist block contains the Best, the 3rd block the Worst, the rest elements are placed in the 2nd block

Best/Worst over Hierarchically Organized Facets

- Consider a facet having a hierarchically organized set of values
- **T ={ European, Asian, BMW, Fiat, Kia, Toyota, Lexus}** organized as follows





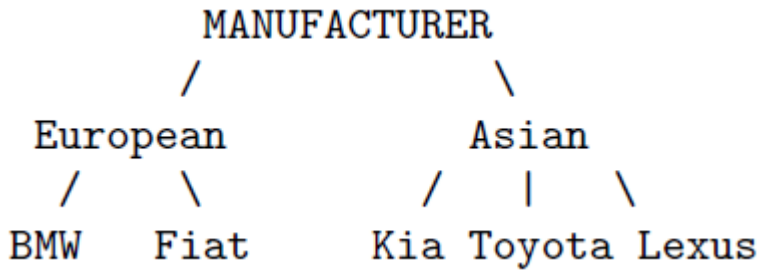
- The non leaf terms can be exploited for **easing** the expressions of preferences (as in faceted search): they are **inherited**!
- **Best(European)** is inherited, so its semantics will contain Best(BMW) and Best(Fiat)
- The induced linear extension will be

  <{European,BMW,Fiat}, {Asian, Kia, Toyota, Lexus}>

Cont.

- The same is done with the **relative** preferences

e.g. **European > Asian**,
actually defines the following
preference relation:



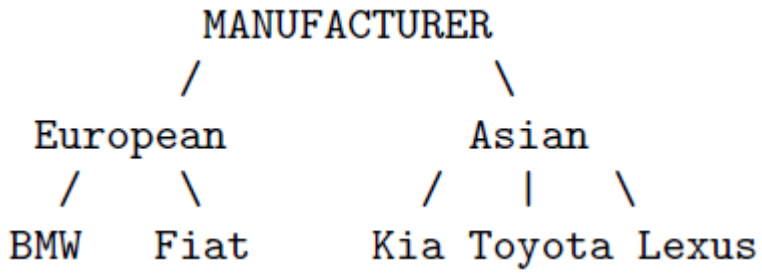
{
European > Asian, European>Kia, European, Toyota, European> Lexus,
BMW> Asian, BMW >Kia, BMW> Toyota, BMW > Lexus,
Fiat> Asian, Fiat >Kia, Fiat> Toyota, Fiat > Lexus
}

- The induced linear order will be

<{European,BMW,Fiat}, {Asian,Kia,Toyota,Lexus}

Cont.

But what should happen if the user provides two "conflicting" preferences, e.g. **European > Asian**, and **Toyota > Fiat**



The first defines the preference relation:

{

European > Asian, European>Kia, European > Toyota, European> Lexus,

BMW> Asian, BMW >Kia, BMW> Toyota, BMW > Lexus,

Fiat> Asian, Fiat >Kia, **Fiat> Toyota**, Fiat > Lexus

}

that contains **Fiat>Toyota** which contradicts the given **Toyota>Fiat**.

Solution: **automatic resolution of the conflict based on scopes**: the ordering of each pair of values is determined by actions having the smallest scope.

So the preference relation will contain **Toyota > Fiat**, it will not contain **Fiat>Toyota**

- Linear Extension: <{European,BMW,Toyota}, {Asian,Kia,Fiat,Lexus}>

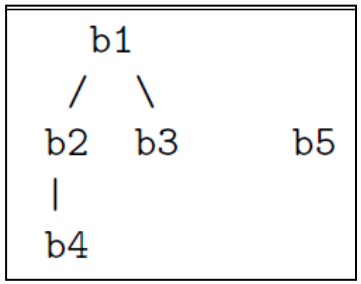
Scope-based Resolution of Conflicts

- A preference relation R over a set of elements E is **valid** iff it is acyclic.
- Each preference action b is **inherited** to its narrower terms. This is the ***scope*** of b.
- We say that an action b is **equally or more refined** than b' iff scope(b) ⊆scope(b').
- *Scope-based Dominance Rule*
  - If A ⊆ scope(b) ⊆ scope(b') then b' is dominated by b on A, and thus action b' should not determine the ordering of A
- The ***active scope*** of b is defined as
  - aScope(b) = scope (b) \ {e ∈ scope(b') | b' is more refined than b}

Example of automatic conflict resolution with 5 conflicting actions

preference	expansion	active scope
b1: $Asian \succ European$		
b2: $European \succ Kia$		
b3: $BMW \succ Asian$		
b4: $Kia \succ Fiat$		
b5: $Toyota \succ Kia$		

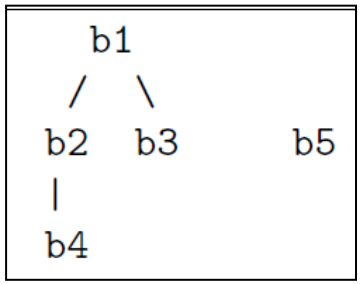
Example of automatic conflict resolution with 5 conflicting actions (cont'd)



Hasse Diagram of the relation *refined* over the actions {b1,..,b5}

preference	expansion	active scope
b1: $Asian \succ European$	$Asian \succ European$, $Asian \succ BMW$, $Asian \succ Fiat$, $Kia \succ European$, $Kia \succ BMW$, $Kia \succ Fiat$, $Toyota \succ European$, $Toyota \succ BMW$, $Toyota \succ Fiat$, $Lexus \succ European$, $Lexus \succ BMW$, $Lexus \succ Fiat$	
b2: $European \succ Kia$	$European \succ Kia$, $BMW \succ Kia$, $Fiat \succ Kia$	
b3: $BMW \succ Asian$	$BMW \succ Asian$, $BMW \succ Kia$, $BMW \succ Toyota$, $BMW \succ Lexus$	
b4: $Kia \succ Fiat$	$Kia \succ Fiat$	
b5: $Toyota \succ Kia$	$Toyota \succ Kia$	

Example of automatic conflict resolution with 5 conflicting actions (cont'd)



Hasse Diagram of the relation *refined*

preference	expansion	active scope
b1: $Asian \succ European$	$Asian \succ European$, $Asian \succ BMW$, $Asian \succ Fiat$, $Kia \succ European$, $Kia \succ BMW$, $Kia \succ Fiat$, $Toyota \succ European$, $Toyota \succ BMW$, $Toyota \succ Fiat$, $Lexus \succ European$, $Lexus \succ BMW$, $Lexus \succ Fiat$	$Asian \succ European$, $Asian \succ Fiat$, $Toyota \succ European$, $Toyota \succ Fiat$, $Lexus \succ European$, $Lexus \succ Fiat$
b2: $European \succ Kia$	$European \succ Kia$, $BMW \succ Kia$, $Fiat \succ Kia$	$European \succ Kia$, $BMW \succ Kia$
b3: $BMW \succ Asian$	$BMW \succ Asian$, $BMW \succ Kia$, $BMW \succ Toyota$, $BMW \succ Lexus$	$BMW \succ Asian$, $BMW \succ Kia$, $BMW \succ Toyota$, $BMW \succ Lexus$
b4: $Kia \succ Fiat$	$Kia \succ Fiat$	$Kia \succ Fiat$
b5: $Toyota \succ Kia$	$Toyota \succ Kia$	$Toyota \succ Kia$

Composing the Preferences of Different Facets

- Actions like those presented so far, lead to the definition of k preference relations, one for each Ti: >1,.., >k.

- They can be composed to define a preference relation over V=T1 x … x Tk (in particular over the Cartesian product of their powersets $\wp$(T1) x … x $\wp$(T1)).

- Two composition methods are proposed: **Pareto** composition and **Prioritized** composition.

Composing the Preferences of Different Facets (cont'd)

Example:

- Sedan >1 Van
- Diesel >2 Gasoline

Obj	Type	Fuel
o1	Sedan	Diesel
o2	Sedan	Gasoline
o3	Van	Diesel
o4	Van	Gasoline

Linear extension of

- >1
  - <{o1,o2},{o3,o4}>
- >2
  - <{o1,o3},{o2,o4}>
- >(1➢2) // >1 has more priority
  - <{o1},{o2},{o3},{o4}>
- >(2➢1) // >2 has more priority
  - <{o1},{o3},{o2},{o4}>
- >(1x2) // equal priority
  - <{o1},{o2,o3},{o4}>

Composing the Preferences of Different Facets (cont'd)

- The Prioritized composition of two preference relations >1 and >2, denoted by >(1➢2) , means that >1 has more priority than >2.
  - It is defined as:
  - (a,b) >(1➢2) (a',b') iff (a >1 a') or ((a=a') and (b >2 b'))

- The Pareto composition of two preference relations >1 and >2 , denoted by >(1x2) , means that both have the same priority.
  - It is defined as follows:
  - (a,b) >(1x2) (a',b') iff
    - (a >1 a') and (b>=2b') OR
    - (b >2 b') and (a>=1a')
  - The first block of the induced ordering it the Pareto optimal set, else called skyline.

(skyline)

- An object o =⟨ v_1, ..., v_K ⟩ **dominates** an object o′= ⟨ v_1', ..., v_K' ⟩ if for each $i \in [1..k]$ it holds $v_i \geq v_i'$ και $\exists\, j \in [1..k]$ such that v_j ‘> v_j

- An object $o \in Obj$ belongs to the **skyline** if it is **not dominated** by another object

Live demo of priorities

- *Script*
  - *Preferences over the terms of two facets*
  - *Switch the order*

The syntax of preference actions

⟨*stmt*⟩ ::= ⟨*scopeType*⟩⟨*spec*⟩

⟨*scopeType*⟩ ::= `facets order :` | `terms order :` | `object order :`

⟨*spec*⟩ ::= ⟨*anchor*⟩⟨*rankSpec*⟩

⟨*anchor*⟩ ::= `facet` ⟨F_i⟩
| `term` ⟨t_j⟩
| `object` ⟨o_k⟩
| ϵ // the empty string

⟨*rankSpec*⟩ ::= {`lexicographic` | `count` | `value` | `indexedBy`} {`min`|`max`}
| `best` | `worst`
| `use scoreFunction` ⟨*score*()⟩ {`min`|`max`}

⟨*stmt*⟩ | `facets order : prefer` ⟨F_i⟩ `to` ⟨F_j⟩

⟨*stmt*⟩ | `terms order : prefer` ⟨t_i⟩ `to` ⟨t_j⟩

⟨*stmt*⟩ | `object order : prefer` ⟨t_i⟩ `to` ⟨t_j⟩

⟨*stmt*⟩ | `objects order : Pareto` ⟨*setOfFacets*⟩

⟨*stmt*⟩ | `objects order : ParetoOptimal` ⟨*setOfFacets*⟩

⟨*stmt*⟩ | `objects order : Priority` ⟨*orderedSetOfFacets*⟩

⟨*stmt*⟩ | `objects order : Combinational` ⟨*bucketOrderedSetOfFacets*⟩

Algorithms

From a Binary Relation to a Bucket Order

- Algorithm **SourceRemoval** (essentially: *Topological sort*)

Algorithm 2 `SourceRemoval`(R)
Input: a binary relation R over E
Output: a bucket order over E that respects R

1: $L \leftarrow \langle\rangle$
2: **repeat**
3: $\quad S \leftarrow maximal_{\succ}(R)$
4: $\quad R \leftarrow R \setminus \{(x \succ y) \in R \mid x \in S\}$ // Remove maximal
5: $\quad L \leftarrow L.append(S)$ // Append a bucket to L
6: **until** $S \neq \varnothing$
7: **return** L

Flat Facets and Preferences

- Algorithm **Apply**
  - Takes as input *Best, Worst* and *Relative Preferences* and produces a bucket order

Algorithm 1 Apply($E, B, W, R_{\succ}, Policy$)
Input: the set of elements E, the set of best elements B, the set of wor
relationships $R_{\succ}$, and $Policy$ for *inactive* elements
Output: a bucket order over E that respects R

1: $R_{bw} \leftarrow \{(b, w) \mid b \in B, w \in W\}$ // each best is preferred than each worst
2: $R \leftarrow R_{bw} \cup R_{\succ}$ //add relative prefs
3: $L \leftarrow$ `SourceRemoval`(R) //produce blocks with boundaries
4: $I \leftarrow E \setminus (B \cup W \cup dom(R_{\succ}))$ // I contains the inactive elements
5: $L' \leftarrow$ `addInactiveElements`$(L, I, Policy)$
6: **return** L'

Hierarchical Facets and Preferences

- Algorithm **PrefOrder**
  - Keypoint: Scope-based resolution of conflicts

Algorithm 4 `PrefOrder`$(E, \mathcal{B}, Policy)$
Input: the set of elements E, the set of actions $\mathcal{B}$, and $Policy$ for *inactive* elements
Output: a bucket order over E

1: *// Part (i): Computation of* $(\mathcal{B}, \sqsubseteq)$ — *Order the actions B based on their scopes*
2: Compute the *scopes* of the actions in $\mathcal{B}$
3: Form $(\mathcal{B}, \sqsubseteq)$
4: *// Part (ii): Efficient Computation of Act. Scopes* — *Compute the active scopes of B*
5: Use $(\mathcal{B}, \sqsubseteq)$ to compute the *active scopes* of the actions in $\mathcal{B}$
6: Use the active scopes to expand the set $\mathcal{B}$ to a set $\mathcal{B}'$
7: *//Part (iii): Derivation of the final bucket order* — *Use the active scopes to unfold the inherited preferences*
8: $(B, W, R_{\succ}) \leftarrow \text{Parse}(\mathcal{B}')$
9: **return** Apply$(E, B, W, R_{\succ}, Policy)$ // call to Alg. 1

Prioritized Composition

- Algorithm **MFOrder**
  - Keypoint:
    - Produce the bucket order defined by the preferences having the highest priority.
    - Then order each block based on the preferences having the 2nd priority, and so on.

Algorithm 6 `MFOrder`$(A, \mathcal{B}_i, \mathcal{B}_j)$

Input: the objects of current focus A, the set of actions $\mathcal{B}_i$ for facet F_i, and the set of actions $\mathcal{B}_j$ for facet F_j

Output: a bucket order of A corresponding to $\mathcal{B}_i \rhd \mathcal{B}_j$

1: We call the Alg. `PrefOrder`$(A, \mathcal{B}_i)$ and let $L = \langle A_1, \ldots A_M \rangle$ be the produced bucket order where M is the number of blocks returned.

2: For each block A_m of L $(1 \leq m \leq M)$ where $|A_m| > 1$, we call `PrefOrder`$(A_m, \mathcal{B}_j)$, returning a bucket order $L_m = \langle A_{m1}, \ldots, A_{mz} \rangle$.

3: We replace each block A_m of L with its bucket order L_m and this yields the final bucket order $L = \langle L_1, \ldots, L_M \rangle$.

Optimized Focus-based Algorithm

- Objective:
  - Define an algorithm whose complexity does not depend on Obj but on A (the current focus).
- Approach
  - Define an algorithm which can be applied to large information bases if the user first restricts (through plain faceted search, i.e. with left clicks) to a focus set A of reasonable size.
  - It is well known that Faceted Search (and consequently PFS) allows restricting very fast the focus set.
    - For instance, the analysis in [Sacco,FQAS'06] shows that **3 zoom operations** on leaf terms are sufficient to reduce an information base of 10^7 objects, described by a taxonomy with 10^3 terms, to an average of 10 objects.
    - A more detailed analysis is availeble in [Tzitzikas, Papadakos, FI'2013]

FS Convergence

- Assume *k* complete and balanced tree taxonomies of **depth** *d* and **degree** *b*. Each leaf indexes **10 objects**
- Optimally the number of **choices** is k ∗ b ∗ d while number of **clicks** is k ∗ d
- Selection of desired **10 objects** from a **peta-sized** collection needs only **30 clicks**
- **Dynamic** taxonomies will further **reduce** the number of **choices**

$n/10$	k	b	d	Num. of Choices $b * d * k$	Num. of Clicks $k * d$
531.441 ($\sim 10^6$)	3	3	4	36	12
3.486.784.401 ($\sim 10^{11}$)	5	3	4	60	20
$\sim 10^{15}$	10	3	3	90	30

Number of facets

Degree of the taxonomies

Depth of the taxonomies

Number of terms the user will see

Number of **decisions**

Optimized Focus-based Algorithm (cont'd)

- How … to avoid Obj
  - We can understand if an action b is more refined than a b' by looking at the anchors of b and b' (i.e. without having to compute their scopes)
  - For each object $a \in A$ we can find the actions to whose active scope a belongs
    - We can understand if $a \in scope(b)$ by looking at the description of a and the anchor of b (i.e. without having to compute $scope(b)$)

Computational Complexity

Table 5. Complexity for non-optimized and optimized Alg. `PrefOrder` and $\texttt{PrefOrder}_{\texttt{Opt}}$

Part	Non-Optimized (Alg. `PrefOrder`)	Optimized (Alg. $\texttt{PrefOrder}_{\texttt{Opt}}$)	Optimized (Alg. $\texttt{PrefOrder}_{\texttt{Opt}}$) for relative prefs
Part 1	$\mathcal{O}(\|Obj\|(\|Obj\| + \|\mathcal{B}\|^2))$	$\mathcal{O}(\|\mathcal{B}\|^2)$	$\mathcal{O}(\|\mathcal{B}\|^2)$
Part 2	$\mathcal{O}(\|Obj\|\|\mathcal{B}\| + exp)$	$\mathcal{O}(\|A\|(\|\mathcal{B}\| + \|\sqsubseteq\|) = \mathcal{O}(\|A\|\|\mathcal{B}\|^2)$	$\mathcal{O}(\|A\|^2\|\mathcal{B}\|^2)$
Part 3	$\mathcal{O}(\|Obj\|^2)$	$\mathcal{O}(\|A\|^2)$	$\mathcal{O}(\|A\|^2)$
Overall	$\mathcal{O}(\|Obj\|(\|Obj\| + \|\mathcal{B}\|^2))$	$\mathcal{O}(\|A\|(\|A\| + \|\mathcal{B}\|^2))$	$\mathcal{O}(\|A\|^2\|\mathcal{B}\|^2)$

No Obj

Set-Valued Facets

- Consider a facet Accessories containing terms like:
  - ABS
  - ESP (Electronic Stability Program}
  - AT (Auto-Transmission)
  - DVD
  - GPS
  - …
- A car can have more than one accessories.
- Consequence for preferences
  - As before, the user can express his/her preferences over each of the accessories, and define a preference relation >i
  - However, since each car can have more than once accessories, we should extend the preference relation >i to a binary relation over P(Ti).

Set-Valued Facets (cont'd)

- Objective
  - We want to order two sets s and s′ of accessories according to our preference
- Approach
  - We count how many elements of s "win" elements of s′ and the other way around. Extra criteria for breaking ties

Def. 3. (Induced Preference over Sets: *MoreWins-Rule*)
If s, s' are two subsets of E, with $wins(s, s')$ we will denote the number of "times" s beats s' according to $\succ$. Formally:

$$wins(s, s') = |\{(e, e') \mid e \in s, e' \in s', e \succ e'\}|$$

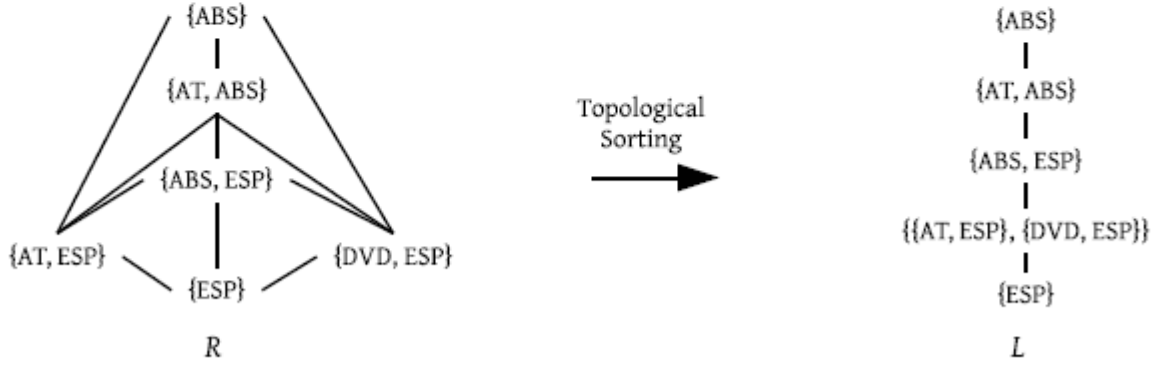
Any subset S of the powerset of E (i.e. $S \subseteq P(E)$), can be ordered according to a preference relation that we will be denoted by $\succ_{\{\}}$, defined by the following rule:

$$s \succ_{\{\}} s' \text{ iff } wins(s, s') > wins(s', s) \quad \square$$

Set-valued Facets (cont'd)

- Let $E = \{ABS, ESP, AT, DVD\}$
- Let ABS be **best**, ESP be **worst** and $ABS \succ AT$
- Let $S = \{\{ABS\}, \{ESP\}, \{ABS, ESP\}, \{AT, ABS\}, \{AT, ESP\}, \{DVD, ESP\}\}$

$w(s, s')/w(s', s)$	{ABS}	{ESP}	{ABS,ESP}	{AT, ABS}	{AT, ESP}	{DVD, ESP}	all
{ABS}	0/0	1/0	1/0	1/0	2/0	2/0	5/0
{ESP}	0/1	0/0	0/1	0/2	0/1	0/1	0/5
{ABS,ESP}	0/1	1/0	1/1	1/2	2/1	2/1	3/2
{AT,ABS}	0/1	2/0	2/1	1/1	3/0	3/0	4/1
{AT,ESP}	0/2	1/0	1/2	0/3	1/1	1/1	1/3
{DVD,ESP}	0/2	1/0	1/2	0/3	1/1	1/1	1/3



- $L = \langle \{ABS\}, \{AT, ABS\}, \{ABS, ESP\}, \{\{AT, ESP\}, \{DVD, ESP\}\}, \{ESP\} \rangle$

Set-valued Facets (cont'd)

- But how to break ties when we compare sets with only **best** or only **worst elements?**

Definition (Breaking ties: *MoreGoodLessBad-rule*)

- If $wins(s, s') = wins(s', s) = 0$ and $Support(s) > Support(s')$ then $s \succ_{\{\}} s'$, where $Support(s) = \sum_{e \in s} sup(e)$ and $sup(e) = |\{e' \in E \mid e \succ e'\}| - 1$

- Let both *ABS* and *ESP* be **best** and *AT* and *DVD* be **worst**

$$wins(\{ABS\}, \{ABS, ESP\}) = wins(\{ABS, ESP\}, \{ABS\}) = 0$$
$$wins(\{AT\}, \{AT, DVD\}) = wins(\{AT, DVD\}, \{AT\}) = 0$$

- Then, $Support(\{ABS, ESP\}) = 2 > Support(\{ABS\}) = 1 > Support(\{AT\}) = -1 > Support(\{AT, DVD\}) = -2$
- So, $L = \langle \{ABS, ESP\}, \{ABS\}, \{AT\}, \{AT, DVD\} \rangle$

Set-Valued Facets (cont'd)

- Algorithm
  - Find the sets of accessories of the cars in A
  - Order these sets according to preferences
  - Use this ordering for ordering the sets in A

- Various optimizations are described in the paper
  - [J. FI 13] Yannis Tzitzikas and Panagiotis Papadakos. Interactive Exploration of Multidimensional and Hierarchical Information Spaces with Real-Time Preference Elicitation. In *Journal FUNDAMENTA INFORMATICAE*, Volume 122, Issue 4, pp 357-399, 2013.

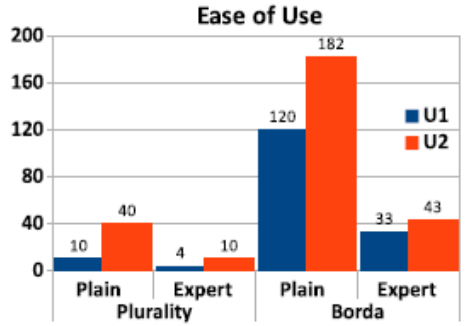
Evaluation of PFS with Users

Evaluation with Users

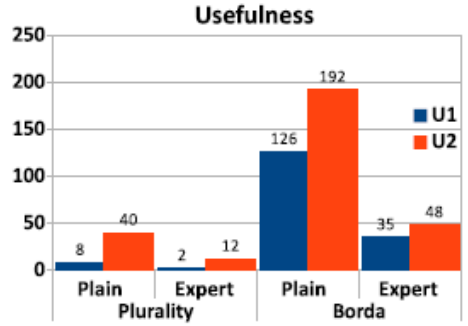
- Evaluate **two** UIs over an information base of 50 cars
  - **plain FDT** UI_1
  - **FDT with preferences** UI_2
- Two user groups
  - **20** participants for **plain** users
  - **6** participants for **expert users**
- **Each user** completed **4 tasks** (2 with UI_1 and 2 UI_2)
  - **2 tasks** required **prioritized composition** and the **rest 2 Pareto**
  - **Plain user** tasks used criteria over **3 attributes**
  - **Expert user** tasks used criteria over **6 attributes**
- **Graeco-Latin Square Design**
  - rotating both the order of tasks and UIs
  - to control order effects

Evaluation: Qualitative Results

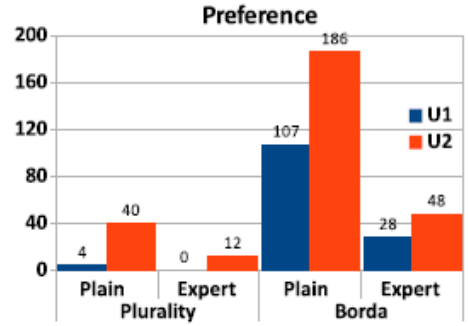
- Users evaluated each UI using a psychometric Likert scale for *Ease of Use, Usefulness, Preference* and *Satisfaction*
- **Main Result:** **All users preferred the PFS over the plain FDT**
  - plain users preferred it:
    - 75% very strongly
    - 20% strongly
    - 5% strong enough
  - expert users preferred it:
    - 50% very strongly
    - 50% strong enough



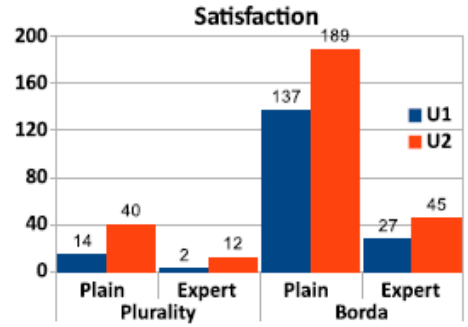
(a)



(b)



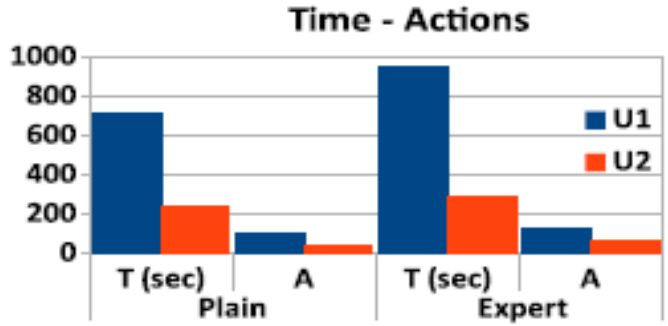
(c)



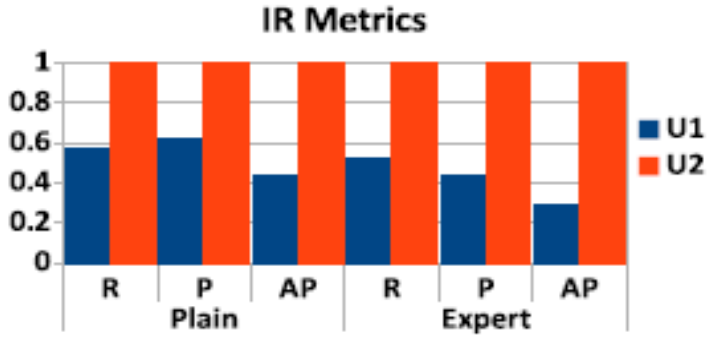
(d)

Evaluation: Quantitative Results

- Users completed **all the tasks successfully** with the **preference** based UI, in **one third of the time** and with **significantly fewer actions!**
- **None of the users** was able to complete **successfully** both tasks with the plain FDT UI
- *Average Precision* is improved **2.30x** for plain and **3.49x** for expert users on average



(a)

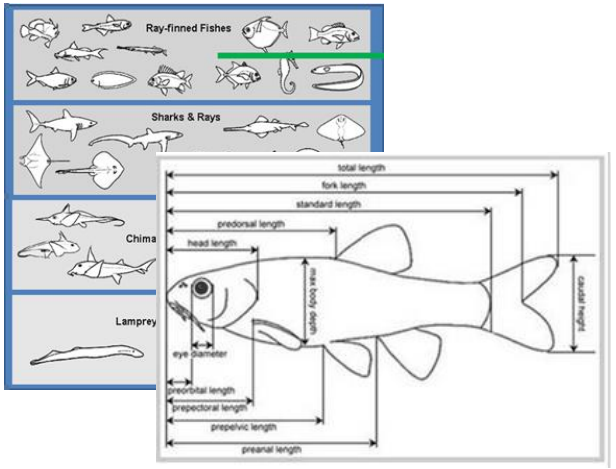


(b)

Other Application Domains

Investigation of Other Application Domains

- For aiding the identification of species
  - Species identification is actually a decision making process comprising steps in which the user makes a selection (or provides as input) that restricts other choices, and so on, until reaching one species. The PFS method offers a flexible process that is order independent.



- As a Voting Advice Application (VAA)
  - In comparison to the widely used questionnaire-based VAAs, the PFS-based method is beneficial with respect to expressiveness, responsiveness, transparency, process and time flexibility.
  - We have developed a pilot appliction for the parliament elections of January 25, 2015 in Greece.

Demos

- Connect with Firefox version 8 or higher:
  - Car Selection : http://62.217.127.128:8080/Hippalus/cars50.jsp
  - Fish Identification: http://62.217.127.128:8080/Hippalus/fishbase300.jsp
  - Voting Advice Application: http://62.217.127.128:8080/Hippalus/parties.jsp
- Express the following preferences:
  - … to be decided in class

Other Related Activities

- Title: Faceted Search with Entity Mining and LOD (Linked Open Data)
  - Facets and terms produced by applying entity mining over the snippets of search hits, where LOD is used as source for entity names
- Motivation
  - LOD contains plenty of information about Named Entities (their names, attributes, relationships with other entities, etc)
- Output
  - IOS Entity Mining
    - LOD is used as source for Named Entity Recognition
    - LOD is used for providing more information about the identified entities

IOS Entity Mining (2012)

http://www.ics.forth.gr/isl/ios

- Automatically connects knowledge with documents at query time
- No preprocessing
- No indexing

Y. Tz... 2014 ...EDB, Athens

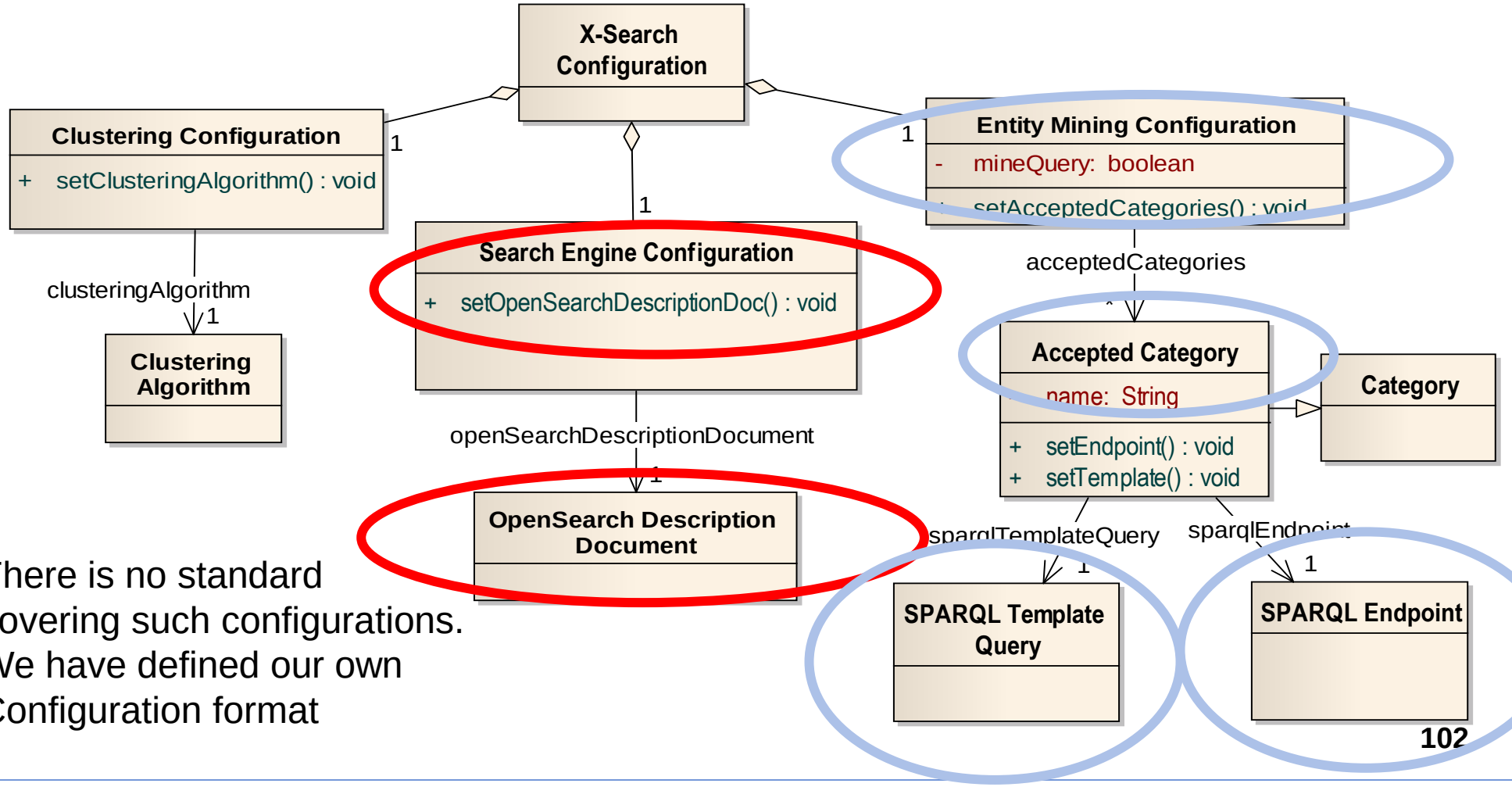
Cont'd

- Exploitation for restricting the focus

Cont'd

- Then we questioned ourselves:
  - Allowing the user to configure himelf the entities of interest by exploiting the LOD
- Outcome
  - X-ENS (e**X**plore **EN**tities in **S**earch)
- Related Publications
  - [SIGIR'13] P. Fafalios and Y. Tzitzikas, X-ENS: Semantic Enrichment of Web Search Results at Real-Time, 36th International ACM SIGIR Conference, Demo Paper, Dublin, Ireland, 28 July - 1 August 2013

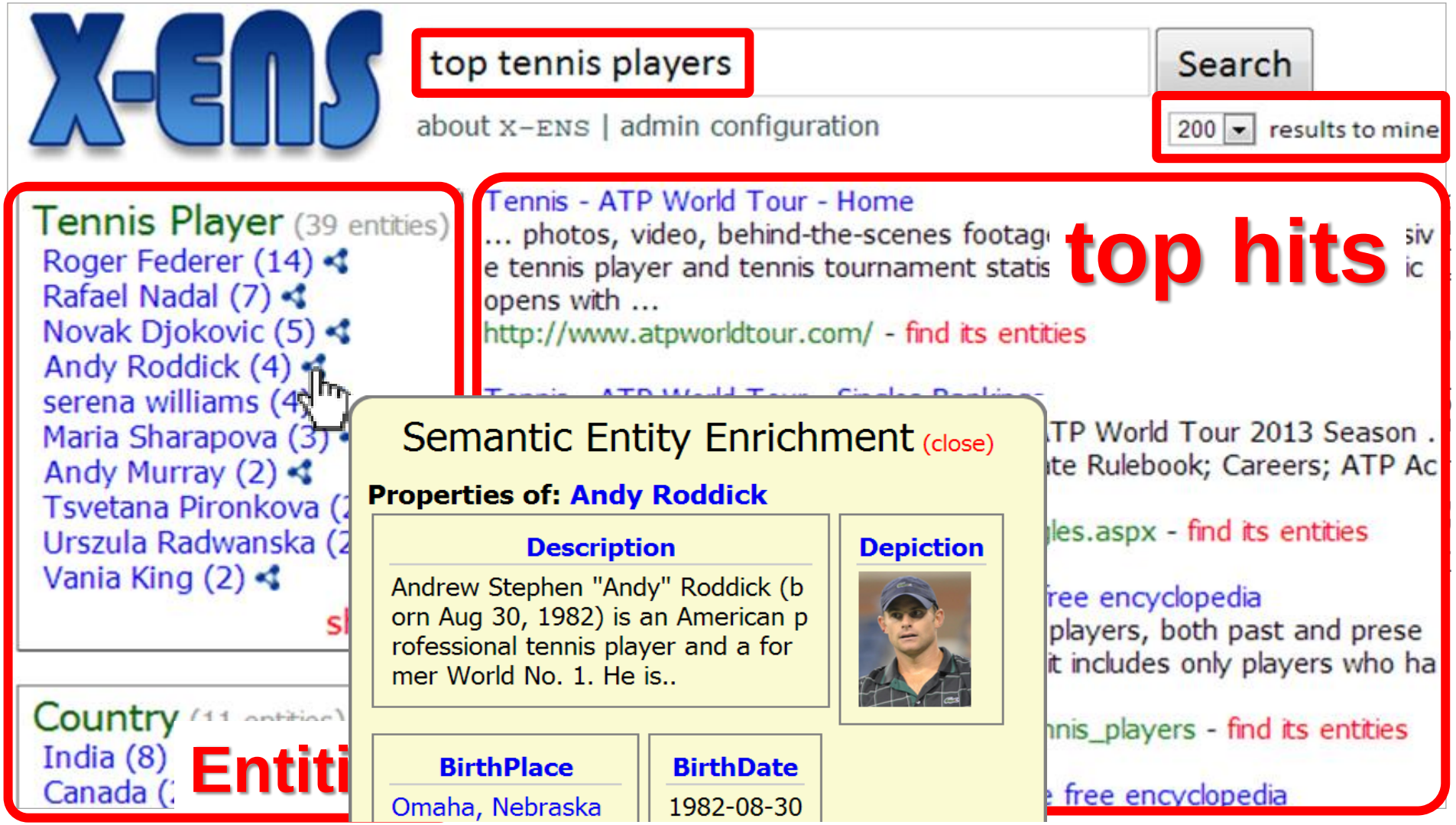
Xsearch-**Configurability**: The Conceptual Model



There is no standard covering such configurations. We have defined our own Configuration format

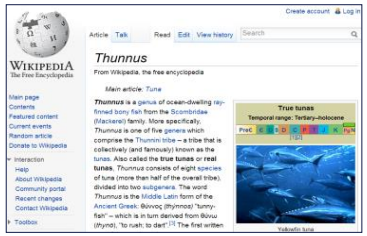
X-ENS (SIGIR 2013)

http://139.91.183.72/x-ens



Xsearch in the context of the operatating iMarine Research Infrastructure: Semantic post-processing of search results

http://www.i-marine.eu/



web browsing

contents

query terms

(top-L) results (+ metadata)

gCube BE Google bing OpenSearch

The LOD cl

MarineTLO Warehouse

Entity Mining

entities / contents

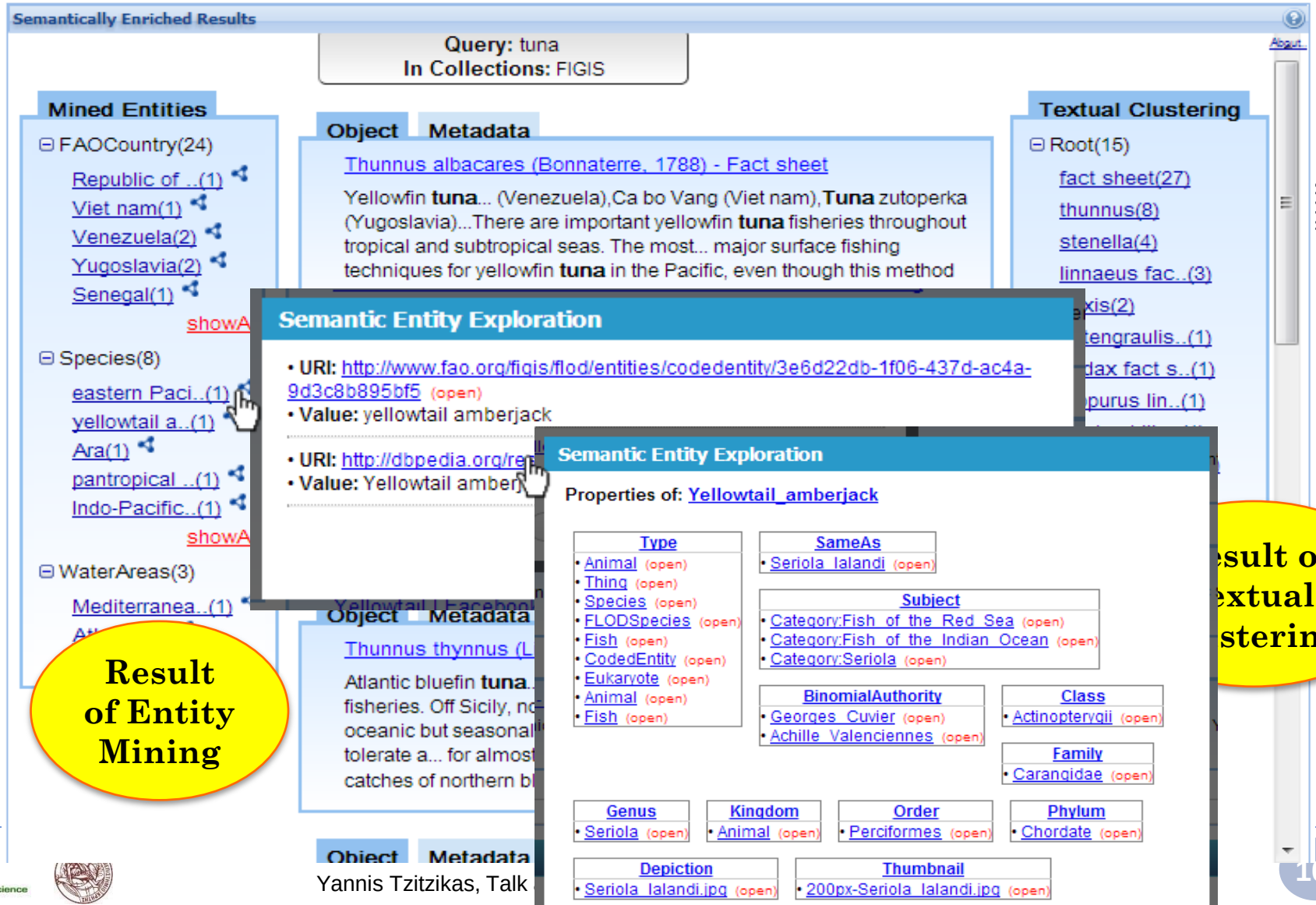
Visualization/Interaction
(faceted search, entity exploration, annotation, top-k graphs, etc.)

semantic data

Semantic Analysis

- Grouping,
- Ranking
- Retrieving more properties

Example: X-Search deployed in an Operational Research Infrastructure (2012-now)



Synopsis

Synopsis

- We have discussed information needs of exploratory nature
- We have seen the basics of faceted exploration
- We have seen an extension of the interaction paradigm of Faceted Exploration with actions that allow the users to express at browsing time their preferences. The user has to kinds of actions:
  - actions that change the focus (zoom-in/out/…) and
  - actions that rank the focus
- The proposed model supports progressive preference elicitation, inherited preferences and automatic scope-based resolution of conflicts over single or multi-valued attributes with hierarchically organized values. Finally we elaborate on the algorithmic perspective and the applicability of the model over large information bases
- The evaluation with users have shown that users completed all the tasks successfully with the preference based UI in one third of the time and with significantly fewer actions.

Future Work and Research

- Investigate requirements stemming from the new applications
  - As a Voting Advice Application
  - As a Species Identification Service
  - Directly over SPARQL results
- Interaction and Algorithms
  - Investigate approaches that rely on fewer preference actions and design faster algorithms for these cases
    - E.g. embed Skyline algorithms for finding fast the first block of the preference-based bucket order

References and Links

References and Links

- **Faceted Search and Dynamic Taxonomies**
  - Sacco, Giovanni Maria; Tzitzikas, Yannis (Eds.), Dynamic Taxonomies and Faceted Search: Theory, Practice, and Experience, Series: The Information Retrieval Series , Vol. 25, 2009
  - Giovanni Maria Sacco: **Dynamic Taxonomies: A Model for Large Information Bases.** IEEE Trans. Knowl. Data Eng. 12(3): 468-479 (2000)
- **Browsing Approach for (plain and fuzzy) RDF**
  - Sébastien Ferré's Publications
  - Nikos Manolis, Yannis Tzitzikas: **Interactive Exploration of Fuzzy RDF Knowledge Bases.** ESWC (1) 2011: 1-16

Cont.

- **Extending Faceted Search with Preferences**
  - [J. FI 13] Yannis Tzitzikas and Panagiotis Papadakos. Interactive Exploration of Multidimensional and Hierarchical Information Spaces with Real-Time Preference Elicitation. In *Journal FUNDAMENTA INFORMATICAE*, Volume 122, Issue 4, pp 357-399, 2013.
  - [ExploreDB'14] Panagiotis Papadakos, Yannis Tzitzikas: Hippalus: Preference-enriched Faceted Exploration. EDBT/ICDT Workshops 2014: 167-172
  - P. Papadakos. Interactive Exploration of Multi-Dimensional Information Spaces with Preference Support . PhD Dissertation, University of Crete, November 2013.
- **Video** Demonstration available at http://www.youtube.com/watch?v=Cah-z7KmlXc
- Links to **Online Prototypes** http://www.ics.forth.gr/isl/Hippalus