A semantic-based fully visual application for matchmaking and query refinement in B2C e-marketplaces

Simona Colucci
Politecnico di Bari
Via Re David, 200 I-70125
BARI, Italy,
s.colucci@poliba.it

Tommaso Di Noia
Politecnico di Bari
Via Re David, 200 I-70125
BARI, Italy,
t.dinoia@poliba.it

Eugenio Di Sciascio
Politecnico di Bari
Via Re David, 200 I-70125
BARI, Italy,
disciascio@poliba.it

Francesco M. Donini
Universitá della Tuscia
Via San Carlo, 32
01100 Viterbo, Italy
donini@unitus.it

Azzurra Ragone
Politecnico di Bari
Via Re David, 200 I-70125
BARI, Italy,
a.ragone@poliba.it

Raffaele Rizzi
Politecnico di Bari
Via Re David, 200 I-70125
BARI, Italy,
raffaele@raffaelerizzi.com

ABSTRACT

This paper presents a visual application in the framework of semantic-enabled e-marketplaces aimed at fully exploiting semantics of supply/demand descriptions in B2C and C2C e-marketplaces. Distinguishing aspects of the framework include logic-based explanation of request results, semantic ranking of matchmaking results, logic-based request refinement. The visual user interface has been designed and implemented to be immediate and simple, and it requires no knowledge of any logic principle to be fully used.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Decision Support; I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages

General Terms

Algorithms, Languages, Economics

Keywords

Semantic Web, Matchmaking, E-marketplaces

1. INTRODUCTION

In this paper we present a web application for semantic discovery and selection of products in a B2C e-marketplace. The main objective we tackle is providing users with benefits of semantic annotation, including richness of descriptions, semantic matchmaking and logic-based ranking and explanation services, while hiding from them all technicalities and letting users experience interaction with the system in an immediate and user friendly way.

The query formulation process is very important for the success of a retrieval system, especially an ontology-based one. The query language has to be very simple for the end user but, at the same time, its expressiveness must be able to capture the real user needs and retrieve only what the user is really looking for. Users are often unable to use logic formulas needed to use a formal ontology [2], they need visual representation to manipulate the domain of interest similarly to what Visual Query Systems do [8]. It is well-known that a challenge for B2C e-marketplaces is to match resources in the e-marketplace to potential buyer's interests, but also to present available goods in an appealing manner, facilitating exploration and selection of product characteristics. As pointed out in [24], selecting a product to buy in e-marketplaces is usually quite a frustrating experience: finding products best fiting users needs and/or financial capabilities often requires too much effort and time, spent browsing web sites or taxonomies in the web sites. Especially when the searched product is not a perfectly defined item, users may have a vague idea of what they are actually looking for, being unaware of all the characteristics of the product. Searching for a product or service often requires domain knowledge that users do not have, so that many potential buyers tend to prefer traditional sales channels, such as physical stores where shop assistants can help the customer to make the right choice and answer to users requests or doubts.

A central issue in e-commerce is hence to support the user in the searching process of the products or services: converting site visitors to buyers in e-commerce environments is a recognized challenging subject [23].

The promise of the Semantic Web is to make information available on the web machine-understandable. By means of formal ontologies, modeled using OWL[21], the knowledge on specific domain can be modeled and exploited in order to make explicit the implicit knowledge, and reason on it thanks to the formal semantics expressed in OWL. Since its launch, the semantic Web initiative has attracted several

ICEC '06, August 14-16, 2006, Fredericton, Canada.

researchers, has raised big money in research projects, has provided many useful insights in knowledge representation, but up to now has provided few working applications.

Obviously, semantic web technologies open extremely interesting new scenarios, including: formalization of annotated descriptions that are machine understandable and interoperable, without being biased by usual drawbacks of natural language expressions; the possibility to reason on descriptions and infer new knowledge; the validity of the Open World Assumption, overcoming limits of structured-data models. There are several reasons for this strange situation: the annotation effort is considerable, though promising results are being obtained on automated extraction and ontology mapping and merging [27]; computational complexity is often demanding also for simple reasoning tasks; interaction with semantic-based systems is often cumbersome and requires skills that most end users do not have –and are not willing to learn.

Furthermore we believe that the effort of annotation should be rewarded with inferences smarter than purely deductive services such as classification and satisfiability, which, although extremely useful show their limits in approximate searches. The question "show us something useful you could not do without semantic web technologies" starts to come up often now.

In this paper we face some of the above issues in the framework of semantic-enabled e-commerce and present an approach and a system, which allow semantic-based matchmaking and retrieval of offers, and query refinement, in an intuitive and propose an extremely intuitive user interface for product discovery.

Main features of our approach are: full exploitation of non-standard inferences for explanation services in the query-retrieval-refinement loop; semantic-based ranking in the request answering; fully graphical and usable interface, which requires no prior knowledge of any logic principles, though fully exploiting it in the back-office. Modeling the marketplace domain using an OWL ontology, the user is able to browse the domain knowledge starting form "her vague idea" on the good she wants to buy. Once the request of the user is formalized with respect to the domain ontology, its formal relations are exploited in order to find goods within the marketplace able to satisfy her needs. Based on the formal semantics of both the request and the returned goods descriptions, an explanation of the matchmaking process results is then provided to the user, who can simply use such new knowledge to refine or change, in a visual environment, her request

The remaining of the paper is structured as follows: next section outlines common sense user needs to be satisfied by an e-marketplace. In Section 3 motivations for a semantic based approach for matchmaking in e-marketplaces are presented together with a brief summary of semantic-based matchmaking in Description Logics. The description of an ontology-based web application satisfying common sense user needs is then presented. Related work and conclusion close the paper.

2. IDENTIFYING COMMON SENSE USER NEEDS

We present, with the aid of real–life example, some of the issues an E-marketplace system should successfully face to help satisfying users needs. Here and in the rest of the paper we focus on an automotive domain and on a car E-marketplace, though the approach is obviously general.

"*Gioseppe has been hired by a new company. He loves the new job, the salary is good. But there is a drawback: the company is in an isolated place, 20 Km far from his city and there is no bus connection. So he needs a car immediately. Then Gioseppe goes to a used car seller, sets his budget, and asks for a car endowed of good safety features – he has to travel up and down for 40 Kms a day – but absolutely the color must not be yellow. He likes Italian style so he preferably would like a car of an Italian manufacturer. Based on these requirements, the shop dealer proposes to Gioseppe some cars he has in stock. While examining proposals, Gioseppe discovers new characteristics he had not asked for, he now considers interesting. Then he reformulates his request and asks for a car of an Italian make endowed of ABS system, airbags, as before excluding yellow color, but now he also would like leather seats and an alarm system. Again the dealer proposes a new set of cars but they do not satisfy Gioseppe so he decides to look also for non-Italian cars, even if he continues to desire an Italian one. He continues refining his requirements until he finally finds, among all the available ones, the car satisfying his needs.*"

The above description illustrates what is the level of interaction humans expect when they interact with other humans. In which ways an automated B2C e-markeplace can provide comparable levels of interaction?

Support to the user in the searching process. Facilitate browsing and selection of product characteristics. The selection of the product really fulfilling user requirements should be the goal of any system for e-commerce. In a real store, users can rely on shop assistants who can help in choosing among various brands or various models. Often a user may not know exactly what she is looking for as she enters a shop, because of lack of specific domain knowledge *e.g.*, she may want to buy a car, but she does not know all the features or optionals a car has.

In the same way, starting from your initial and vague idea on what you wish to buy, you may go around supermarket aisles and then find the product that best fits your wishes: a user in an e-marketplace could "discover" some new unknown product characteristics searching for the ones she already knew. A system should support users in the searching process starting from incomplete information they have on the product to be bought/sold. It should manage incomplete information in the user requirements and also in the products descriptions. The (potential) buyer, especially in the initial stage, might be not aware of all the possible characteristics she can specify for a particular product. As said before, this may happen both because the user could be not a domain expert and because she has not an "exact" idea on what she wants to buy. On the other hand, the individual who describes the items in the marketplace decides the characteristics she wants to emphasize in the description.

The system should support users in the elicitation of their needs, in order to refine the initial query and reflect their true needs or wishes. One of the main problems with *preference elicitation* is that preferences expressed by user on the initial stage of the search process can be uncertain and erroneous so preference elicitation process has to be part of the searching process, as preferences can depend on partial search results [2]. Moreover the process of eliciting preferences should not require neither much effort on the user's side or force him to set a large number of weights on items. It is well-assessed that assigning a precise and explicit value (weight) to each item, especially when the number of items increase, is extremely frustrating for users.

Efficiency and trust . A key issue for a system supporting users in e-marketplace is not simply finding *a* product, but finding the *right* product[24], or a set of products, best matching the user needs. Furthermore users should be *confident* that the system has made the best choice for them. A simple presentation of a list of items after a query may be not sufficient to convince the user they are the best choice. In the same way a user may not be convinced they are proposed because of their match degree with the request or because some other "unclear" factor. Explanations on match degree may help in both cases.

Ranking criteria . Often products in e-marketplaces, or in general in Internet shopping malls, are compared only through their *price*. The most common tools for product selection over internet sites present offers as an ordered list, sorted according to their price, or more generally in increasing order of a quantitative attribute. The limit is here the possibility to choose each time only one criterion to display the results [23]. Nevertheless, price is not the only characteristic to be considered in the request unless the item has been perfectly identified. Users preferences are expressed over a set of attributes describing the good to be bought. Considering only one criterion is not realistic. For instance, an E-marketplace supporting the sale of cars has to model different features, as look, comfort, optionals, model and not only the price, quantity or delivery time. So a system should be able to provide overall rankings according to users requests.

Friendliness . an e-marketplace system should not need any specific skill or learning effort to be immediately usable also by non expert-users. Experience in information retrieval shows that users may encounter problems even with simple text-based Boolean expressions, by far preferring graphical query interfaces [2]. [1]

3. WHY ONTOLOGIES?

Currently, the approaches adopted by web-based tools in order to retrieve resources within a repository – as an e-marketplace can be seen as a particular repository – are mostly either database oriented or text retrieval based, while a minority of them uses taxonomies. Why are they not enough? What do they miss with respect to the user needs introduced in the previous section? Database systems are extremely efficient in handling huge amounts of data (items). The state of the art in database systems allows the management of very big marketplaces whose items expose many characteristics. A drawback of such an approach is in its "closed world" assumption. What is not specified is considered as a constraint of absence, as a negation by default approach is used. It is not hence possible to manage incomplete information. Using a database is also not possible to have explanations on the results presented. Using text retrieval based techniques, well known problems of noise and bad recall have to be taken into account [3]. It is difficult to find the best match and if some heuristics are used to refine the results, the system does not present any explanation on them to clarify system behavior. Taxonomies are very useful to browse classes of items. Each node in a taxonomy can represent a set of items sharing a common characteristic. But, once this initial set of items has been found, it is not possible to use the taxonomy to refine the query.

Besides the above mentioned limitations, all these approaches lack of the possibility to deal with the semantics of the descriptions – both the user request and resources descriptions; a very useful feature in the search process. In taxonomy-based approaches a very basic semantic search (IS-A relation between category in the tree) is presented, but it results very weak. We believe that especially in e-marketplaces, the "meaning" of the terms rather than the terms themselves is very important. Turning back to Gioseppe, if he was looking for a `safe car`, then a car endowed with `ABS system` and `airbags` would be a good choice. In order to catch these logical correlation, ontologies [17] would help **Gioseppe** in the search process. An ontology allows to relate terms with each other and give a formal model to the knowledge of the marketplace domain, and consequently express that a `safe car` is a `car` `endowed_with` an `ABS_system` and `endowed_with` `airbags`.

Exploiting the formal semantics of the language used to build the ontology, logic based inference processes can be performed, successfully dealing also with incomplete information (Open World Assumption – OWA). Based on such inference services an efficient retrieval process can be carried out.

Nevertheless, using standard deductive inference services only exact matches can be identified. Neither logical ranking nor explanation services on resources discarded during the search process are available, as the reasoning engine behaves as a boolean oracle. As in database systems, a list of results is presented to the user apparently without any justification.

To overcome such limitations, in [12] an ontology based approach has been proposed exploiting abductive inference services and belief revision techniques [11] in a Description Logics based framework. Using these services both explanation on the results can be provided to the user and new knowledge elicitation can be performed in order to guide the user in her query refinement.

3.1 Semantic Based Matchmaking

A close relation exists between OWL and Description Logics. In fact, the formal semantics of OWL DL sub-language is grounded in the Description Logics theoretical studies.

For the sake of completeness, we briefly recall standard inference services in Description Logics (DLs) and Concept Abduction and Concept Contraction services, showing how they can be used in a matchmaking process for match explanations and knowledge elicitation. We assume the reader be familiar with the basics of Description Logics [1].

The appendix provides anyway, for the interested reader–

[1] In this paper we are not specifically interested in an accurate usability analysis and do not perform any usability tests, which is part of ongoing work.

a brief guided tour of Description Logics.

3.1.1 Match Classes.

Given an ontology $\mathcal{T}$ and two DLs formulas D (for demand) and S (for supply), two standard inference services are provided by a DL reasoner:

Subsumption : Check if S is more specific than D with respect to the ontology $\mathcal{T}$. In a formal way: $\mathcal{T} \models \mathsf{S} \sqsubseteq \mathsf{D}$.

Satisfiablity : Check if S (conversely D) is satisfiable with respect to the ontology $\mathcal{T}$. In a formal way: $\mathcal{T} \not\models \mathsf{S} \sqsubseteq \bot$.

Based on these standard inferences, given a request D (for demand) and a resource S (for supply) the following match classes can be identified with respect to an ontology $\mathcal{T}$(see [14, 22, 18]).

exact $\mathcal{T} \models \mathsf{D} \equiv \mathsf{S}$. S *is semantically equivalent to* D. All the characteristics expressed in D are presented in S and S does not expose any additional characteristic with respect to D.

full $\mathcal{T} \models \mathsf{S} \sqsubseteq \mathsf{D}$. S *is more specific than* D. All the characteristics expressed in D are provided by S and S exposes also other characteristics both not required by D and not in conflict with the ones in D.

plug-in $\mathcal{T} \models \mathsf{D} \sqsubseteq \mathsf{S}$. D *is more specific than* S. All the characteristics expressed in S are provided by D and D requires also other characteristics both not exposed by S and not in conflict with the ones in S.

potential $\mathcal{T} \not\models \mathsf{S} \sqcap \mathsf{D} \sqsubseteq \bot$. D *is compatible with* S. Nothing in D is logically in conflict with anything in S.

partial $\mathcal{T} \models \mathsf{S} \sqcap \mathsf{D} \sqsubseteq \bot$. D *is not compatible with* S. Something in D is logically in conflict with some characteristic in S.

Actually, it is questionable whether a **plug-in** match type should be considered better than **potential** one. We note that some researchers also consider **plug-in** match more favorable than **full** match (*e.g.*, see [18, 22]), motivating this choice with the idea that if D *is more specific than* S one may expect that the advertiser offering resource S will probably have also more specific resources; in an e-commerce setting if the advertiser offers a *sedan* car it will also probably offer specific types of *sedans*.

Nevertheless we argue that this idea prevents a fully automated matchmaking, which is possible when S *is more specific than* D, and furthermore it favors underspecified resource description, *i.e.*, an advertisement offering a sedan will always plug-in match any request for a specific sedan, but will leave on the requester the burden to determine the right one – if any is actually available – for her needs. Even though exact match is surely the best match, full match might be considered –not always anyway– equivalent from the requester point of view, because it states that at least all the features specified in D are also expressed in S. We can give a rank to the match classes:

partial $\rightarrow$ **potential** $\rightarrow$ **full** $\rightarrow$ **exact**

Largest part of logic-based approaches only allow, as pointed out before, a categorization within match types.

3.1.2 Non Standard Inference Services for Logical Matchmaking.

Notice that even if **exact** and **full** matches are obviously the best possible matches, in resource retrieval scenarios the most frequent cases are **potential** and **partial** matches. We can evaluate a score for **potential** and **partial** matches considering their distance from a **full** match and explain the match degree proposing: (a) in case of **partial** match, which characteristics have to be retracted from D in order to reach a **potential** match with S; (b) in case of **potential** match, what is not specified[2] in S in order to be more specific than $\mathsf{D}$and then have a **full** match. The knowledge elicitation for query refinement can be performed evaluating what is exposed in S and is not required by D. In order to perform these evaluation two non-standard inference services for DLs can be exploited. We briefly recall them.

Given two DLs formulas D and S both satisfiable with respect to an ontology $\mathcal{T}$

Concept Contraction : If D and S are not compatible with each other –$\mathcal{T} \models \mathsf{D} \sqcap \mathsf{S} \sqsubseteq \bot$– find two DLs formulas G (for give up) and K (for keep), such that both $\mathcal{T} \models \mathsf{D} \equiv \mathsf{G} \sqcap \mathsf{K}$ and $\mathcal{T} \not\models \mathsf{K} \sqcap \mathsf{S} \sqsubseteq \bot$.

Concept Abduction : If S is not more specific than D –$\mathcal{T} \not\models \mathsf{S} \sqsubseteq \mathsf{D}$– find a formula H (for hypotheses) satisfiable with respect to $\mathcal{T}$and such that $\mathcal{T} \models \mathsf{S} \sqcap \mathsf{H} \sqsubseteq \mathsf{D}$.

(partial $\rightarrow$ potential) Hence, if D and S are in a **partial** match, solving a Concept Contraction it is possible to compute G representing why D is in conflict with S and K representing the new contracted request. After the contraction we have K in potential match with S.

(potential $\rightarrow$ full) If D and S are in **potential** match, solving a Concept Abduction problem, hypotheses H on why there is not a **full** match between D and S are computed. Hence the conjunction $\mathsf{S} \sqcap \mathsf{H}$ is a **full** match for D.

Using Concept Contraction and Concept Abductions is then possible to compute and explain how far is a **partial** match or a **potential** match from a **full** match.

If D and S are in potential match, the characteristics B (for bonus)[13] specified in S but not requested in D represent the knowledge that can be elicited and proposed to the requester in order to still refine the initial query. At this point it should be easy to see how B can be computed solving a Concept Abduction problem.

In [12] we proposed a formalization on how to deal with **strict** and **negotiable** characteristics, in a DL framework. Roughly speaking, the demander sets a characteristic as strict, if she never wants to give up that strict feature. Then all the resources that are in a partial match because of the strict constraints have to be discarded (see [12] for further details). If all the request characteristics are set strict, then only potential matches are allowed. This case models the situation where the user is not willing to contract any part of her request in order to reach a potential match within the marketplace. Typically in a refinement iterative process this is the case when the user formulate her initial query. In fact, if she is not satisfied by the system results to her first query, then she starts to negotiate on some constraints in order to find appealing offers.

[2] We write "not specified" instead of "missing" in order to emphasize the underlying Open World Assumption.

4. I WOULD LIKE TO BUY A CAR, BUT...

Based on the theoretical framework presented in the previous sections we developed a (Java webstart)web application[3] fully exploiting semantic-based matchmaking procedures and aimed at satisfying user needs identified in Section 2. The application is Ontology –and marketplace– independent, *i.e.*, the visual interface is built on the fly once the reference ontology –hence the chosen marketplace domain– has been selected (Figure 1).

Figure 1: Marketplace/Ontology Selection

Then, if the selected marketplace is changed, the new ontology can be loaded to dynamically present the new available domain knowledge to the user. The GUI main panel is a visual representation of the knowledge modeled by the ontology. Actually, the effectiveness of how the information is visualized is strongly dependent on the quality of the ontology. As pointed out also in [15], it is very difficult to manage the visualization of a poorly structured ontology.
Note that here we do not deal with marketplace population. Once the ontology has been selected, the corresponding marketplace and all its semantically annotated supplies become available for the discovery process.

4.1 The Tool

When the application starts, the GUI main panel appears divided in two main sections: the left-side one (Figure 2(a)(b)(d)), from now on navigation panel, is devoted to ontology browsing – intensional navigation [9], the right-side one (Figure 2(c)(e)), from now on query panel, to graphically visualize the user request. The **navigation panel** is further divided in two panels. In the leftmost one (Figure 2(a)) the entry-points for the intensional navigation are represented in the top side. The navigational style is *top-down* fashion. Initial characteristics the user can select in order to model her request are only the most generic ones within the ontology – that is, all those classes which are direct children of the <owl:Thing/> class. By selecting one of these classes, both all its sub-classes and the roles having the selected class as domain are visualized within the right side of the visualization panel (Figure 2(b)). The user starts from a general aspect of the domain and then is guided in depth with recursive zooms on the ontological model.

Clicking on one of these just visualized characteristics, the intensional navigation continues recursively exploiting the sub-class or the domain/range relations and the new information is visualized within the same panel. What the user sees in the navigation panel are local views of the ontology. Doing so the user is allowed to concentrate on the current focus of her search in the right side of the navigation panel and can easily change at will the entry point, if she decides to look for something different. To help the user in the navigation and to come back to an upper level – zoom out – a history bar is visualized on the navigation panel (Figure 2(d)). The information of the visualized characteristic within the GUI, is not just the name of either the class or the role within the ontology. Exploiting <rdfs:comment/> meta-information within an OWL file we can associate a describing image to each class and role name and show it as an icon of the class/role. Moreover <rdfs:label/> tag is used to manage multilingual information.

If the user finds a characteristic she is interested in, she drags it in the **query panel** and adds it to her initial request. If the dragged element is a <owl:SameClassAs/>, then its definition is added to the query panel. The query panel is divided in two sections, so that the user can visually express both positive and negative preferences. In the top side (Figure 2(c))the user drags characteristics she would like the retrieved supplies have, in the bottom side (Figure 2(e)) the user drags the characteristics she explicitly does not want in the retrieved supplies. We recall that under an OWA the negative information must be clearly stated. The user is only aware of atomic characteristics to be added or removed from the query panel. All the logical relations between these characteristics, which are coded within the ontology, are completely hidden to her. As the user can add elements to the query panel, she can remove them just right-clicking on the corresponding item. It is noteworthy that in the initial query all the characteristics are set strict *i.e.*, mandatory.

Once the user formulates her request, the selection process is performed matchmaking her request with all the supplies semantic-enabled descriptions available in the marketplace using an external reasoning engine, exploiting the formal semantics of both the request and supplies descriptions.

The reasoner is obviously not embedded within the web application. This one communicates with the inference engine via a DIG 1.1 interface over HTTP. Since the tool exploits both standard and non-standard inference services recalled in Section 3.1 we use MaMaS-tng[4] reasoning engine, which exposes a standard DIG 1.1 interface enhanced with additional tags to support the above mentioned services.

Selection results are shown in a **results window**. The information within this panel are twofold: a *semantically-ranked list* of available supplies – with respect to the query – within the marketplace; *explanations* on the match results and *suggestions* on how to refine the query adding new characteristics found in the retrieved offers but not specified in the user request (see [12] for the logical rationale of this process). The results window is shown in Figure 3: in the left side – **list panel**, the ranked list endowed with match explanation for each retrieved supply is presented (Figure

[3] http://sisinflab.poliba.it/marketplace/

[4] http://sisinflab.poliba.it/MAMAS-tng/

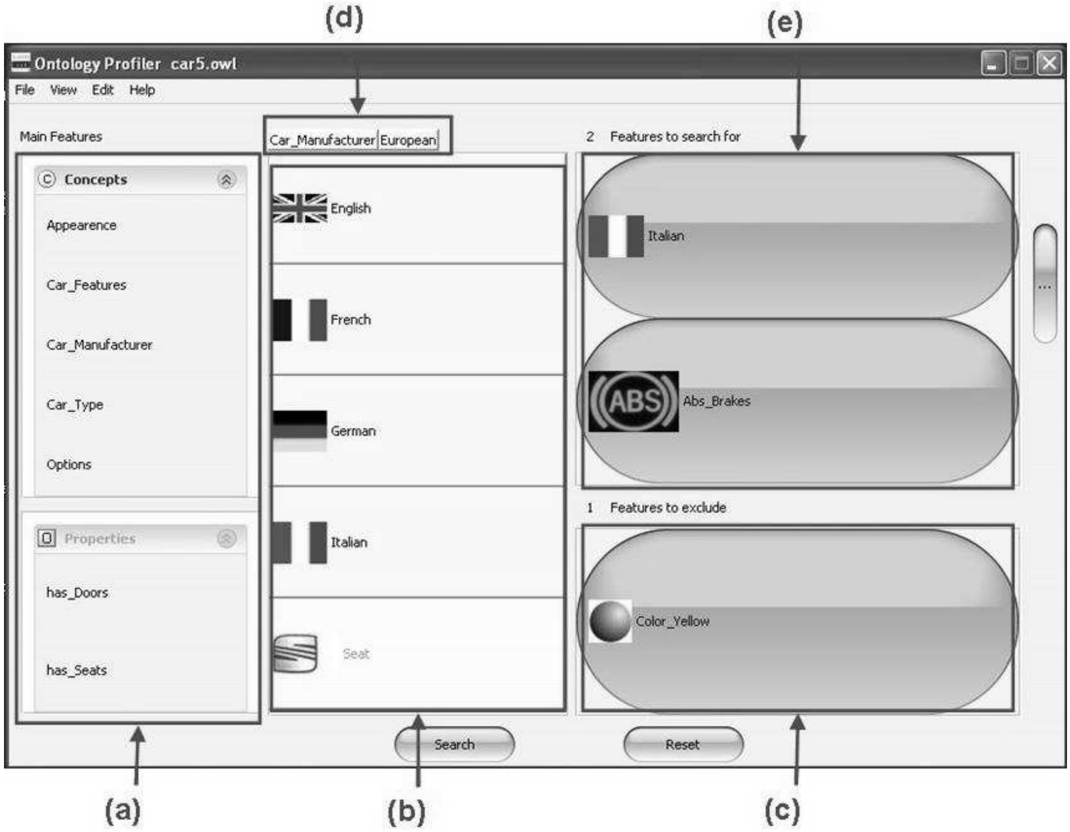

Figure 2: The Graphical User Interface for intensional navigation and query formulation

3(a)(b)); in the right side – **query refinement panel**, a visualization of the query is presented in the top side (Figure 3(c)) and suggestions on characteristics to be added to the query at will in the bottom side (Figure 3(d)).
In the **list panel**, for each retrieved item, the system provides the information detailed hereafter:

description An image representing the retrieved item together with a natural language description of the item itself. The system provides also a transliteration of the semantic-based supply description. The verbalization of the OWL description is provided automatically by the system [9].

match value Based on the semantics of both the query and the offers descriptions, a semantic similarity value is computed [12]. This value is used to rank the results list.

match explanation A semantic based explanation for the match result is displayed. Fulfilled, unspecified, conflicting and additional characteristics, with respect to the request, are displayed for each item. Obviously, if all the the query characteristics are set to strict, the conflicting elements set is empty and is not displayed. Additional features represent what is not specified in the query but is specified in the offer.

The list panel has a multi-page visualization (Figure 3(b)). For each page only five items are displayed.

The **query refinement panel** is divided in two sections: in the top side panel the query is visualized, in the bottom side all the additional information – bonus – related to the offers visualized in the current page of the list panel, are displayed (Figure 3(d)). The query refinement panel allows the user to refine the query in two different ways: relaxing some characteristics setting them to negotiable or adding new characteristics from the additional ones of the currently displayed supplies. If the user sets the characteristic to negotiable (gray colored features in the top side panel – Figure 4), then also the supplies exposing a characteristic in conflict with the negotiable ones are taken into account during the refinement stage. Notice that setting characteristics to negotiable is not equivalent to removing them from the query. It is not a *don't care* specification. A negotiable characteristic has to be interpreted as a *wish* specification. That is why, also supplies in conflict with the negotiable features are considered during the matchmaking process and ranked in the final result list. The bonus characteristics in the query refinement panel represent information the user might not be aware of or she initially considers not relevant for the search. Nevertheless, it is related to what the user is looking for. If the user asks for a `sedan` and the retrieved supplies expose among additional features the `air conditioning`, then the user could be interested in this bonus exposed by some sedan cars.
Once the user adds new features to the query selecting them from the additional ones or she sets some characteristics to negotiable, a new search can start based on the new refined

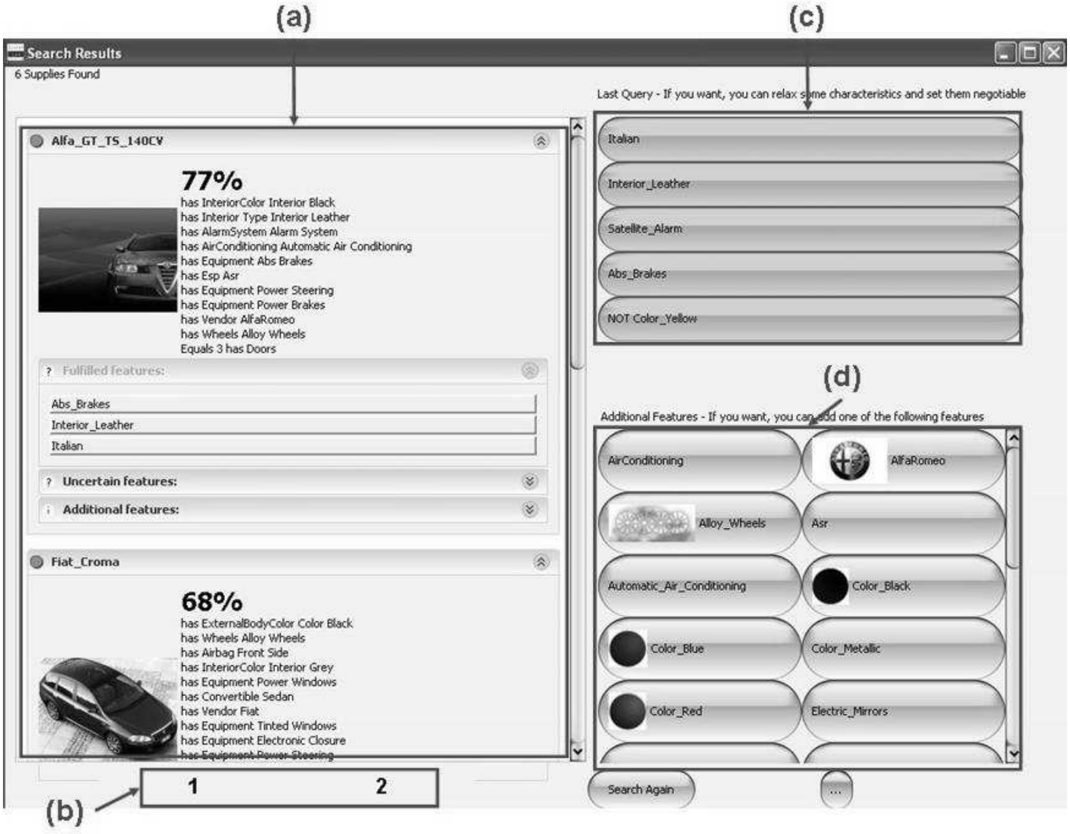

Figure 3: The results window

query. The process can be iterated until the user finds a supply best fitting her needs.

4.2 The backoffice

In this section we describe the matchmaking process, *i.e.*, what happens behind the scenes during the search process after the query (re-)formulation.

1. Thanks to the intensional navigation the user formalizes her request with respect to the ontology. In this initial query all the requested characteristics are set strict.

2. All the supplies within the marketplace compatible with the request, *i.e.*, in potential match with it, are retrieved. For each of them, solving concept abduction problems via `MaMaS`-tng, fulfilled, uncertain and additional feature are computed based on the knowledge modeled within the ontology. They represent respectively: which part of the request is also present in the supply description; what is requested by the user but is not specified in the supply description; the bonus characteristics. Based on fulfilled and uncertain characteristics, a semantic-based match score is computed.

3. All the retrieved supplies are ranked with respect to their semantic-based match score and then grouped in sets of five elements each. The first group of supplies is displayed in the list panel. All the additional features related to these supplies are put together and displayed in the bottom side of the query refinement panel. If the user selects another group/page (Figure 3(b)) then the bottom side of the query refinement panel is updated with the bonuses related to the new displayed supply.

4. If the user does not find any supply satisfying her needs, she can refine the query.

5. After the query refinement, a new retrieval process is carried out. All the supplies in conflict with the new strict characteristics are discarded. For the remaining supplies, if they are in partial match with the query then, for each of them, contraction problems are solved to compute both a contracted request (see Section 3.1.2) which is in potential match with the supply and conflicting features. Concept abduction problems are then solved in order to compute fulfilled, uncertain an bonus characteristics with respect to the contracted request. Based on conflicting, fulfilled and uncertain characteristics the semantic-based match score is computed.

6. The process restarts form point 3.

4.3 User needs satisfaction

Turning back to the user requirements and needs outlined in Section 2, we now explain how the proposed application tries to satisfy them.

Support to the user in the searching process. Using intensional navigation, the user is guided through the ex-

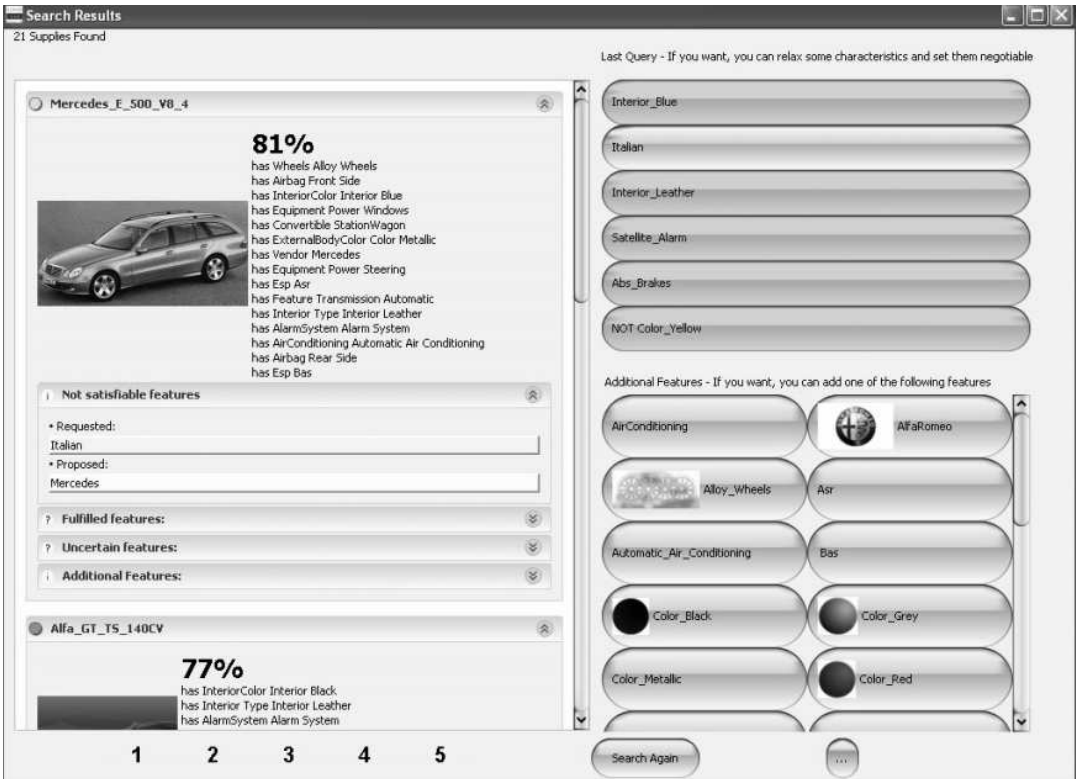

Figure 4: The results window after a query refinement

ploration of the marketplace knowledge domain. Even if she is not a expert, she can start from very general concepts of the domain and discover what she is really looking for. Furthermore, since the user only sees local views of the ontology, she can focus only on it in each step of the query formulation.

The preference elicitation is managed by the tool using bonus characteristics. In fact since only the additional information related to the current page in the list panel is shown, then if the user is interested in one the supplies presented in that page, probably she is also interested in their bonus features.

Giving the opportunity to set negotiable some characteristics in the query, the user is helped in expressing also her wishes – "*preferably, I would like*".

Efficiency and trust. All the retrieved supplies are selected considering user's strict specifications and needs. Exploiting the semantics of the request all the supplies in conflict with the user strict requirements are discarded. For the retrieved supplies, the rank is compute based on the meaning of their description and their degree of request satisfaction.

For each retrieved supply, an explanation on the match degree is shown to the user. Then she can explicitly verify why the system chose a supply rather than another one, without having to blindly trust the system, thus increasing her confidence in next interactions with the marketplace system.

Ranking criteria. The ranking is established based on the semantic similarity of the demand with each supply. Of course, the semantic match degree value can be combined with other extra information, *e.g.*, price, quantity or delivery time, in order to refine the ranking function, which can be imported for each marketplace.

5. RELATED WORK

5.1 Example-based tools

Widespread web-based systems for browsing and retrieval of products usually adopt lists, ranked or not, to present results. Obviously, *ranked lists* are preferable to unordered sets of items, nevertheless they become less effective and usable as the complexity of the product description increases, together with possible user preferences. To improve search and visualization various *example-based* search tools have been developed, such as SmartClient [28]. SmartClient uses constraint satisfaction techniques, allows to refine (critique) preference values specified in the first step of the search and supports trade-off analysis among different attributes, *e.g.*, looking for an apartment a user can make a compromise between distance and rent (more distant less expensive). Also in [19] a candidate/critiques model has been presented, which allows users to refine candidate solutions proposed. Here, preferences are elicited incrementally by analyzing critiques through subsequent iterations. It is an *Automated Travel Assistant* (ATA) for planning airline travels, and similarly to SmartClient, ATA exploits CSP techniques: preferences are described using soft constraints defined on the values of attributes. AptDecision [25] is a tool supporting elicitation of preferences in the real estate domain: browsing the domain, users can discover new features of interest and through their refinement of apartment features, agents can build a profile of their preferences using learning techniques. In [26] Conversional Case Base Reasoning (CCBR) is exploited. The tool is dubbed ExpertClerk as it tries to reproduce the way of acting of a human salesclerk: asking questions to understand buyer's needs (navigation-by-asking) and then proposing solutions achieving such needs (navigation-by-proposing). The cycle is repeated until the shopper finds an appropriate product. A drawback of this approach is that it implies an adequate user domain knowledge; users could not know the answers to certain questions and then the interaction can take a wrong direction if the user gives a wrong answer.

ExpertClerk was inspired by FindMe [6], which used case-based reasoning as a way of recommending products in e-commerce catalogs. FindMe, and its enhanced version The Wasabi Personal Shopper (WSP) [5], combined instance-based browsing and tweaking by difference. Different FindMe systems have been developed, in various domains. Among systemsbased on FindMe the most renowned is *Entrée* [4], a restaurant recommender, which allows users to refine a query on the basis of the results displayed, so it is possible to choose a restaurant less expensive or closer than the restaurant shown after the first query.

5.2 Semantic-based tools

Recently, there has been a growing interest towards systems supporting semantics exploitation, in different domains. In [16] an application is presented, improving traditional web searching using semantic web technologies: two Semantic Search applications are presented, running on an application framework called TAP, which provides a set of simple mechanisms for sites to publish data onto the Semantic Web and for applications to consume these data via a query interface called GetData. The results provided by the system are then compared with traditional text search results of Google. Story Fountain [20] is an ontology-based tool, which provides a guided exploration of digital stories using a reasoning engine for the selection and organization of resources. Story Fountain provides support for six different exploration facilities to aid users engaged in exploration process. The system is being used by the tour guides at Bletchley Park. The approach has been further investigated in [10]. An intelligent query interface exploiting an ontology-based search engine is presented in [9]; the system enables access to data sources through an integrated ontology and supports a user in formulating a query even in the case of ignorance of the vocabulary of the underlying information system.

We do not present here related work on semantic matchmaking. The interested reader is referred to [12].

6. CONCLUSION

In this contribution we have presented a system that, in our humble opinion, clearly shows the benefits of semantic markup of descriptions in an e-marketplace. Exploiting ontologies and non-standard inference services for OWL DL ontologies, it allows to satisfy common sense user needs during the interaction within an e-marketplace.

The user is guided in the query formulation through the intensional navigation of a specific marketplace domain knowledge without any underlying knowledge of the so called semantic web technologies. The semantics of the ontology-based query and supplies descriptions is used to perform a semantic-based matchmaking process and to provide explanations on match results.

We are carrying out preliminary tests on the system, with the aid of human volunteers. The domain we selected was one of used cars and first results are extremely encouraging. Experiments are devoted to evaluate both the theoretical approach and the usability of the application. The evaluation of different match degree functions, combining extra-ontological information, is also under investigation.

We finally wish to point out that we chose not to provide a running example of the system behaviour, to encourage readers' interaction with our application, available at http://sisinflab.poliba.it/marketplace/.

Acknowledgments

We would like to gratefully acknowledge the partial support of projects Apulia Region Explorative project ACAB-C2, PON PITAGORA and Apulia Region Strategic project DIPIS.

7. REFERENCES

[1] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2002.

[2] A. Bernstein, E. Kaufmann, A. Göhring, and C. Kiefer. Querying ontologies: A controlled english interface for end-users. In *proc. of 4th International Semantic Web Conference (ISWC 2005)*, pages 112–126, November 2005.

[3] D. C. Blair and M. E. Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3):289–299, 1985.

[4] R. Burke. Knowledge-based recommender systems. In A. Kent, editor, *Encyclopedia of Library and Information Systems*, volume 69. New York, 2000.

[5] R. D. Burke. The wasabi personal shopper: A case-based recommender system. In *Proc. of AAAI/IAAI '99*, pages 844–849, 1999.

[6] R. D. Burke, K. J. Hammond, and B. C. Young. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997.

[7] D. Calvanese, G. De Giacomo, and M. Lenzerini. On the Decidability of Query Containment under Constraints. pages 149–158, 1998.

[8] T. Catarci, M. F. Costabile, S. Levialdi, and C. Batini. Visual query systems for databases: A survey. *journal of visual languages and computing (JVLC)*, 8(2):215–260, 1997.

[9] T. Catarci, P. Dongilli, T. D. Mascio, E. Franconi, G. Santucci, and S. Tessaris. An ontology based visual tool for query formulation support. In *Proceedings of the 16th European Conference onArtificial Intelligence (ECAI '04)*, pages 308–312, 2004.

[10] T. Collins, P. Mulholland, and Z. Zdráhal. Semantic browsing of digital collections. In *proc. of 4th International Semantic Web Conference (ISWC 2005)*, pages 127–141, 2005.

[11] S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. A uniform tableaux-based method

for concept abduction and contraction in description logics. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI '04)*, pages 975–976. IOS Press, 2004.

[12] S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace. *Electronic Commerce Research and Applications*, 4(4):345–361, 2005.

[13] S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and A. Ragone. Knowledge elicitation for query refinement in a semantic-enabled e-marketplace. In *7th International Conference on Electronic Commerce ICEC 05 ACM Press*, pages 685–691. ACM, 2005.

[14] T. Di Noia, F. M. Di Sciascio, Eugenio andDonini, and M. Mongiello. A system for principled matchmaking in anelectronic marketplace. *International Journal of Electronic Commerce*, 8(4):9–37, 2004.

[15] E. Franconi, S. Tessaris, T. Catarci, T. Di Mascio, G. Santucci, and G. Vetere. Specification of the ontology design tool. Technical report, D6.2, SEWASIE Consortium, 2003.

[16] R. V. Guha, R. McCool, and E. Miller. Semantic search. In *Proceedings of the Twelfth International World Wide Web Conference, WWW2003*, pages 700–709, 2003.

[17] I. Horrocks. Owl: A description logic based ontology language. In *ICLP*, pages 1–4, 2005.

[18] L. Li and I. Horrocks. A Software Framework for Matchmaking Based on Semantic Web Technology. *International Journal of Electronic Commerce*, 8(4), 2004.

[19] G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The automated travel assistant. In *Proceedings of the Sixth International Conference on User Modeling*, pages 67–78, Vienna, 1997.

[20] P. Mulholland, T. Collins, and Z. Zdráhal. Story fountain: intelligent support for story research and exploration. In *Proc. of Intelligent User Interfaces Conf.*, pages 62–69, 2004.

[21] OWL. Web Ontology Language. www.w3.org/TR/owl-features/, 2004.

[22] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic Matching of Web Services Capabilities. In *proc. of International Semantic Web Conference (ISWC 2002)*, number 2342 in LNCS. 2002.

[23] P. H. Z. Pu and P. Kumar. Evaluating example-based search tools. In *Proc. of ACM Conference on Electronic Commerce*, pages 208–217, 2004.

[24] G. M. Sacco. The intelligent e-store: Easy interactive product selection and comparison. In *Proc. of IEEE Conf. on Electronic Commerce (CEC '05)*, pages 240–248, 2005.

[25] S. Shearin and H. Lieberman. Intelligent profiling by example. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001)*, pages 145–151. ACM press.

[26] H. Shimazu. Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. *Artificial Intelligence Review*, 18(3-4):223–244, 2002.

[27] P. Shvaiko and J. Euzenat. A Survey of Schema-based Matching Approaches. *Journal on Data Semantics*, 4, 2005.

[28] M. Torrens, B. Faltings, and P. Pu. Smartclients: Constraint satisfaction as a paradigm for scaleable intelligent information systems. *Constraints*, 7(1):49–69, 2002.

APPENDIX

Description Logics (DLs) are a family of knowledge representation languages for knowledge representation. They derive their name by the fact that they focus on concept descriptions, *i.e.*, expressions involving basically atomic concepts and roles.

In Description Logics, basic elements are concept names, role names and individual names. Intuitively, they represents respectively sets of elements in the knowledge domain, binary relations between elements in different concepts, special names for elements belonging to a concept.

More formally, a semantic *interpretation* is a pair $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, which consists of the *domain* Δ and the *interpretation function* $\cdot^{\mathcal{I}}$, which maps every concept to a subset of Δ, every role to a subset of $\Delta \times \Delta$, and every individual to an element of Δ. We assume that different individuals are mapped to different elements of Δ, *i.e.*, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ for individuals $a \neq b$. This restriction is usually called *Unique Name Assumption* (UNA).

Concepts, roles and individual names can be combined to form complex descriptions – concept expressions. For this purpose, each DL has a defined set of logical operator. Every DL allows the use of boolean conjunction, which is interpreted as set intersection, usually denoted as $\sqcap$– $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$. Some DLs include also disjunction $\sqcup$ and complement $\neg$ to close concept expressions under boolean operations whose interpretation is the usual set-theoretic of union and complement. Roles can be combined with concepts using *existential role quantification*, *e.g.*, $\texttt{sedan} \sqcap \exists \texttt{has\_security\_systems.ABS}$ which describes the set of sedans with an ABS system on board, and *universal role quantification*, *e.g.*, $\texttt{sedan} \sqcap \forall \texttt{has\_color.Metallic\_Color}$, which describes sedans having exclusively a metallic color. Other constructs may involve counting, as number restrictions: $\texttt{sedan} \sqcap \leq 3 \texttt{has\_doors}$ expresses sedans with at most three doors, and $\texttt{sedan} \sqcap \geq 2 \texttt{has\_airbags}$ describes sedans equipped with at least two airbags. The interpretation of constructs involving quantification on roles needs to make domain elements explicit: for example, $(\forall R.C)^{\mathcal{I}} = \{d_1 \in \Delta \mid \forall d_2 \in \Delta : (d_1, d_2) \in R^{\mathcal{I}} \rightarrow\ d_2 \in C^{\mathcal{I}}\}$. Many other constructs can be defined, increasing the expressive power of the DL, up to n-ary relations [7].

Concept expressions can be used in *inclusion assertions*, and *definitions*, which impose restrictions on possible interpretations according to the knowledge elicited for a given domain. For example, we could impose that transmission can be divided into automatic and manual using the two inclusions: $\texttt{Transmission\_Type} \sqsubseteq \texttt{Automatic} \sqcup \texttt{Manual}$ and $\texttt{Automatic} \sqsubseteq \neg\texttt{Manual}$.

Or, that secure cars have at least an ABS system as $\texttt{Secure\_Car} \sqsubseteq \exists.\texttt{has\_security\_systems.ABS}$. Definitions are useful to give a meaningful name to particular combinations, as in $\texttt{Italian\_Car} \equiv \texttt{Car} \sqcap \forall \texttt{has\_manufacturer.Italian}$.

Historically, sets of such inclusions are called TBox (Terminological Box). In simple DLs, only a concept name can appear on the left-hand side of an inclusion. The semantics of inclusions and definitions is based on set containment: an interpretation $\mathcal{I}$ satisfies an inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and it satisfies a definition $C = D$ when $C^{\mathcal{I}} = D^{\mathcal{I}}$. A *model* of a TBox $\mathcal{T}$ is an interpretation satisfying all inclusions and definitions of $\mathcal{T}$.