Ontology-driven car pooling via semantic matchmaking: a context-aware approach

Abstract

Despite remarkable progress, location-based discovery of transportation services is currently hindered by the superficial information content associated to resources. Dynamic entities like vehicles in car pooling and fleet management scenarios are characterized only by basic properties such as geographical coordinates, name and a general category. This allows to support only exact matches, often leading to poor user satisfaction. The paper proposes a framework based on knowledge representation languages and Semantic Web technologies to annotate OpenStreetMap resources with articulate information endowed with formal meaning, so supporting semantic-based matchmaking to filter and rank resources according to relevance. The approach was assessed in a case study on an Intelligent Transport System (ITS) enabling advanced discovery, scheduling and tracking of car pooling services.

1 Introduction and motivation

In most developed countries, mobile and ubiquitous services are a big business and an everyday experience. Location-based, context-aware solutions are having a deep impact on the discovery and fruition of several types of resources and services, from Points Of Interest (POIs) on personal navigation devices to real-time on-demand transportation networks. Car sharing and pooling systems are particularly complex organisms involving distributed coordination and matchmaking between providers and clients of transport services. Recent successful enterprises (such as Uber\(^1\), Lyft\(^2\) and BlaBlaCar\(^3\)) are deeply transforming traditional business-making in this sector by exploiting the strategic benefits of information technology and mobile computing devices to facilitate matches among component actors in real time. Almost anyone can quickly join the network and become a provider of payed transport services, without complex management issues. Similarly, consumers can quickly submit a request on their mobile devices and see the best service options on a map, with accurate location and timing information. People are willing to make this kind of sharing economy grow very fast, because they have been accustomed to a peer-to-peer, crowd-sourcing engagement mindset by the World Wide Web.

Nevertheless, technological limitations stifle further progress. A significant hurdle lies in the unrefined information content associated to location-based resources. Current solutions mostly model just coordinates, a name and a category, so supporting only exact matches and coarse “yes-no” outcomes. Retrieval systems cannot exploit articulate descriptions to filter out and rank resources according to relevance with respect to user’s request or profile. The problem is even more evident when dealing with dynamic resources such as public and private transport vectors, which change continuously not only their position but also availability and quality of service parameters. Exposing resources in real time to on-the-go clients is not enough: it is necessary to model them in more meaningful and structured ways, so as to support advanced discovery. This will lead to higher personalization and user satisfaction. At the same time, when dealing with rich information content, attention has to be paid to prevent cognitive overload and complicate interactions: the system should shield the users from the inner complexities of data models, formalisms and algorithms.

\(^1\)https://www.uber.com/
\(^2\)https://www.lyft.com/
\(^3\)https://www.blablacar.com
Knowledge Representation (KR) technologies can enable more detailed and structured descriptions of resources. Borrowing basic ideas from the Semantic Web paradigm for resources on the WWW, the use of metadata (annotations) with machine-understandable meaning can enable advanced location- and context-aware resource discovery through proper automated inferences [18]. Within such perspective, this paper introduces a comprehensive framework for requester/provider matchmaking in car pooling and fleet management services. It leverages Semantic Web technologies to enhance the data model of crowd-sourced OpenStreetMap cartography and to annotate dynamic resources such as vehicles with semantic metadata. Starting from an annotated user profile, the system executes matchmaking with transport resources in a reference region near the user’s location. Both passenger and driver profiles are expressed in Web Ontology Language (OWL) expressions, formally grounded on the Description Logics (DLs) family of KR languages. Non-standard inference services allow to support both exact and approximated matches, also ranking resources by semantic distance, measured formally in terms of conflicting and/or missing features. The feasibility of the proposal was assessed by means of a case study for car pooling and sharing. In the case study, multiple dynamic constraints by drivers and prospective passengers are annotated and matched using an optimized mobile reasoner [17].

The remainder of the paper is structured as follows. Section 2 discusses related work. Section 3 describes in detail the overall proposed framework, while the case study is presented in Section 4. Section 5 closes the paper.

2 Related work

Greater interoperability and flexibility can be achieved in location-aware service discovery by means of ontology-based information annotation and management [1]. Nevertheless, providing paradigms and techniques that are effective yet intuitive enough is still largely an open issue. The route planning framework introduced in [11] used an ontology to describe road segments, modeling both user preferences and context. Though remarkable, the approach requires a single universal ontology, ensuing manageability and extensibility issues. In [12], a prototypical mobile client was presented for semantic-based mobile service discovery. An adaptive graph-based representation allowed ontology browsing. However, usability and performance issues emerged and heuristic mechanisms were needed to simplify interaction.

Research in advanced location-based services has benefited from the free and open cartographic corpuses created in latest years by community projects. The most mature one is, by far, OpenStreetMap (OSM). It provides map data in an open, well-documented format under the Open Data Commons Open Database License, granting anyone the permission to use, copy, share and modify them. In [2] a framework was proposed to transform and publish OSM data as a Linked Data (LD) [3] repository in RDF format. A SPARQL RDF query language endpoint was also exposed, allowing to retrieve geo-data of a given region, optionally filtered by property values.

The position paper [13] suggested the Semantic Web is an enabling technology to overcome the challenges of the next generation of geo-spatial mobile discovery and Augmented Reality (AR) solutions. The work proposed here answers several of the outlined questions. The iPhone client in [19] was able to retrieve an RDF data set relevant to locations and objects in the direction of the user. It integrated open LD sources of cultural heritage collections, but the applicability of the approach was limited by the availability of suitable data sets, since the problem of creating and maintaining them was not considered. The system in [4] exploited an ontology devised to model the evolution of a POI from temporal, thematic (i.e., function) and geometrical (i.e., building structure) viewpoints. Unfortunately, discovery was based only on the thematic class; more refined queries were not possible to filter and rank resources based on specific features.

\[^4\]http://www.openstreetmap.org
\[^7\]SPARQL 1.1 Overview, W3C Recommendation 21 March 2013, http://www.w3.org/TR/sparql11-overview/
Car sharing and pooling has attracted significant research effort in latest years, not only due to social and economic relevance, but also to the complexity of the problem. In [20] a pre-matching approach was adopted: the satisfaction of multiple requests with available vehicle resources was formulated as an integer multiple commodity network flow problem. Unfortunately, this approach does not support dynamic matchmaking and planning. Furthermore, it can take into account only numerical or enumerated data types, while higher level descriptions are not supported. Static optimization was achieved also in [9]: the proposed system first clustered users by proximity, then computed routes. Personal preferences were taken into account, but only as numeric corrective factors on the global optimization function. Other approaches include stochastic modeling [21] and ant colony optimization [8]. These proposals, however, lack support for dynamic resource discovery. Conversely, the mobile Android system in [10] supported dynamic car pooling. Relevant features included a route matching algorithm and a driver reputation system, but issues like matching personal preferences were left open. Semantic-based approaches allow greater flexibility [5, 7]: the one proposed here distinguishes itself from other solutions in the support for complex resource descriptions, approximated matches and resource ranking.

3 Car sharing via semantic matchmaking: proposed approach

In the proposed framework, both users (also named passengers, clients or requesters from now on) and vehicles (also named drivers or resources) are characterized by an OWL compressed annotation [16], which describes relevant properties according to a reference ontology. Users’ features include their relevant preferences (e.g., air conditioning, car radio), transport requirements (e.g., car trunk capacity, vehicle type, baby seat) and potential incompatibility constraints with fellow passengers (e.g., talkativeness, smoking habits, age). Resource descriptions will contain vehicle specifications along with equipment information and the remaining space availability, in addition to driver and trip characteristics.

The overall framework is depicted in Figure 1. It includes:

– Semantic matchmaker. The integrated lightweight reasoner in [17] computes the semantic-based matchmaking and resource ranking. Basic theory is shortly recalled in Section 3.2 to make the approach understandable, while the reader is referred to [14, 17] for wider discussion and algorithm details.

– Map data files. They encapsulate both geographical data and semantic annotations of resources. Such information is dynamically updated according to established car-passenger agreements and trip updates.

– OSM data parser. It extracts semantic annotations from OSM map data for the user local area. They are decompressed and passed to the matchmaker, which performs inference services on the user’s mobile device.

The semantic-based matchmaking engine allows a smart allocation process aiming at maximizing vehicle carrying capacity, also reducing the environmental pollution from traffic. In order to take into account passenger-passenger and passenger-vehicle constraints, non-monotonic inferences [14] are exploited. They automatically detect the best associations by taking into account semantic compatibility degree of passengers among them and with vehicles. Next section describes the elements of the devised framework in greater detail.

3.1 Semantic enrichment of OSM maps

OpenStreetMap data adhere to a simple XML schema with three basic elements: (i) nodes, i.e., single georeferenced points; (ii) ways, ordered sequences of nodes; (iii) relations, grouping multiple nodes and/or ways. Each element includes a unique identification code, latitude and longitude coordinates, versioning information and optional general-purpose informative tags. A tag is a key-value pair of Unicode strings of up to 255 characters. In order to allow drivers annotate both vehicle features and context information (e.g., preferred passengers profiles, trip details), new tags have been introduced [15] complying with the basic OSM tag structure, hence retaining backward compatibility:
The semantic prefix is used to distinguish semantic annotations from other tags. The index \( n \) identifies different annotations—possibly referring to different ontologies—associated to the same map element. Key name suffix and value format differ for each proposed tag type, as in what follows:

- \(<\text{tag k="semantic:n:ontology" v="URI" }/>\) denotes the ontology the semantic node annotation refers to. The tag value is the unique ontology URI (Uniform Resource Identifier), which usually consists of a URL (Uniform Resource Locator) which can be accessed to retrieve the ontology as a file.

- \(<\text{tag k="semantic:n:encoding" v="format" }/>\) specifies the compression format used to encode the semantic annotation. Compression techniques are needed in order to cope with the well-known verbosity of XML-based ontological languages such as RDF and OWL [16].

- \(<\text{tag k="semantic:n:counter" v="data" }/>\) tags contain the Base64 string representation of the compressed semantic annotation. If its length is within 255 characters, a single tag is used, else it is split in 255-character segments and each one is stored in a tag. The \( \text{counter} \) suffix is assigned as a segment index, starting from 1.

In the proposed approach, each vehicle is annotated as an OSM node, added to OSM server after the driver decided to share his private vehicle with other people. He can provide a detailed description of the car by means of JOSM\(^8\) plugin [15], which allows editing semantic annotations through a fully visual user interface, based on simple drag-and-drop operations, as shown in Figure 2. Afterwards, while the car is picking up or dropping off passengers, the OSM semantic annotation is periodically updated with the contextual information based on each passenger profile. Update operations exploit a mobile client with a simple prototypical user interface, similar to BlaBlaCar\(^9\). Only the car owner (e.g., the driver) is allowed to edit the corresponding OSM node. The proposed system allows to monitor and annotate the actual trip, by automatically updating OSM node coordinates.

### 3.2 Knowledge-based passenger/car matchmaking

The discovery framework is based on interest profiles modeling each user’s preferences. They are semantic annotations in OWL 2 language expressed w.r.t. a reference ontology, exploited to perform semantic-based non-standard reasoning procedures. An ontology (a.k.a. TBox, terminological box) is a formal specification of a conceptualization modeling a domain of discourse. In DLs, basic elements

\(^8\)http://josm.openstreetmap.de  
\(^9\)https://www.blablacar.com
include: *concepts* (a.k.a. *classes*), which represent sets of objects; *roles* a.k.a. *object properties*, relationships linking pairs of objects; *individuals* (a.k.a. *instances*), which are particular named objects. They are combined through *constructors* to form *concept expressions*. The *conjunction* constructor \((\sqcap)\) is common to all DLs; some DLs have also one or more of the following: *disjunction* \((\sqcup)\); *negation* \((\neg)\); *existential* \((\exists)\), *universal* \((\forall)\) and *number restrictions* \((\geq n, \leq n)\) on properties, and others.

With more constructors the expressivity of a DL grows, but also the computational complexity of inference (a.k.a. *reasoning*) services. Reasoning is needed to derive implicit consequences from an explicitly modeled *Knowledge Base*, i.e., the union of an ontology and facts about individuals.

Figure 3 shows a relevant excerpt of the ontology modeled for the car pooling domain. Upper-level classes include:

- *Car_Type*, representing the different kinds of cars;
- *Feature*, describing information concerning requested/provided vehicle equipments;
- *Preference*, whose subclasses specify passengers’ or drivers’ personal requirements;
- *Luggage*, which contains classes related to passengers load.

Typical DL-based matchmaking systems adopt standard reasoning services such as *Subsumption* and *Satisfiability*. Given a request \(R\) and an available resource \(S\), Subsumption checks whether all features in \(R\) are included in \(S\): its outcome is either *full match* or not. On the other hand, Satisfiability verifies whether any constraint in \(R\) contradicts some specification in \(S\), hence it divides resources in compatible (a.k.a. *potential matches*) and incompatible (a.k.a. *partial matches*) w.r.t. a given request [14]. This approach, however, usually gives poor results in advanced scenarios, because full matches are rare and incompatibility often occurs when matching articulate descriptions. Standard inferences do not let one understand what constraints caused incompatibility (or prevented a full match), nor how much they are truly important for the user. In order to achieve a principled and granular ranking of potential and partial matches, as well as an explanation of outcomes, *Concept Abduction* and *Concept Contraction* non-standard inference services were adapted from their original e-commerce scenarios [6] to transportation service retrieval. Given a request \(R\) incompatible with an available resource \(S\), Contraction detects what part \(G\) (for Give up) of \(R\) is conflicting with \(S\). If one retracts \(G\) from \(R\), \(K\) (for Keep) is obtained, which represents a contracted version of the original request, such that it is compatible with \(S\). Therefore Contraction works as an extension and an explanation of (un)satisfiability. Conversely, if \(R\) and \(S\) are compatible but \(S\) does not fully satisfy \(R\), then Abduction identifies what is missing in \(S\) in order to reach a full match. In other words, Abduction provides an explanation for (missed) subsumption, returning what additional feature set \(H\) (for *Hypothesis*) should be assumed in \(S\). Furthermore, *_penalty functions* can be associated to both \(G\) and \(H\), in order
to compute a semantic distance of each available resource w.r.t. a given request [6, 14]. In Abduction and Contraction, penalty grows proportionally to the number and complexity of concepts in $H$ and $G$, respectively.

Adopting the above theoretical framework and inference procedures, a score can be assigned to each vehicle (playing the role of resource), expressing the result of matchmaking with the user profile (i.e., the request). Furthermore, resource and request OSM annotations can contain extra-logical features, in order to take into account data-oriented, context-dependent constraints. The semantic distance score should be combined with contextual attributes to yield a global match metric. In the proposed framework, heading of the passenger is compared with the one of the vehicle as a pre-filtering step. Only resources forming an angle below a predetermined threshold will be admitted to the subsequent (more computationally expensive) semantic matchmaking. Furthermore, after matchmaking the overall resource score is computed using the following utility function:

$$u(R, C) = 100\left[1 - \frac{s_{penalty}(R, C)}{s_{penalty}(R, ⊤)} \left(1 + \frac{distance(R, C)}{max\_distance}\right)\right]$$

where $s_{penalty}(R, C)$ is the semantic distance between profile $R$ and car annotation $C$; this value is normalized dividing by $s_{penalty}(R, ⊤)$, which is the distance between $R$ and the universal concept $⊤$ (a.k.a. Top or Thing) and depends only on the ontology structure. Geographical distance (normalized by a proper range threshold) is combined as weighting factor. The purposes of the utility function are to weight the result of semantic matchmaking according to the car distance and to convert the score to a more user-friendly $[0, 100]$ ascending scale. Nearer resources are preferred, but in case of a full match $s_{penalty}(R, C) = 0$ hence $u(R, C) = 100$ regardless of distance. Given a request $R$ and a set of resources $C = \{C_1, C_2, \ldots, C_n\}$, the resource $C_i$ with the highest $u$ score will be selected. In this way each passenger will be allocated to the car that best satisfies her transportation requirements.

4 Case study

In order to better explain the proposed framework and illustrate its benefits, a practical example from a car pooling case study is described. A heterogeneous group of people are sending requests from different places and at different times to the car pooling service in order to reach their destinations.
Let us also consider five drivers currently available for ride sharing through the service with their cars: a SUV, a luxury sedan, two utility cars and a convertible. Figure 4 summarizes relevant geographical data related to users and vehicles (position and heading). Each resource (i.e., vehicle + driver) is endowed with a semantic annotation split –from a logical point of view– into two parts. The first one is constant over time and summarizes both the set of vehicle features relevant to the service and the driver’s profile. The second part specifies information about the space available for passengers and their luggage. Table 1 summarizes the relevant features of each car/driver pair by means of icons (green ones denote the availability of a feature, red ones the absence of it).

![Figure 4: Positions and destinations of users and vehicles](image)

<table>
<thead>
<tr>
<th>Vehicle type</th>
<th>Available space</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="SUV icon" /></td>
<td><img src="image" alt="SUV features icon" /></td>
<td><img src="image" alt="SUV features" /></td>
</tr>
<tr>
<td><img src="image" alt="Sedan icon" /></td>
<td><img src="image" alt="Sedan features icon" /></td>
<td><img src="image" alt="Sedan features" /></td>
</tr>
<tr>
<td><img src="image" alt="Utility 1 icon" /></td>
<td><img src="image" alt="Utility 1 features icon" /></td>
<td><img src="image" alt="Utility 1 features" /></td>
</tr>
<tr>
<td><img src="image" alt="Utility 2 icon" /></td>
<td><img src="image" alt="Utility 2 features icon" /></td>
<td><img src="image" alt="Utility 2 features" /></td>
</tr>
<tr>
<td><img src="image" alt="Convertible icon" /></td>
<td><img src="image" alt="Convertible features icon" /></td>
<td><img src="image" alt="Convertible features" /></td>
</tr>
</tbody>
</table>

Table 1: Vehicle features in the reference example

The problem consists in matching every passenger with the car better fulfilling his requirements; it is also important to consider compatibility issues and preferences among users. For each car, the following information is retrieved through the service: unique identifier of the reference ontology, semantic-based annotation in compressed OWL format and trip information, expressed with geographic coordinates in OSM data structures. Both passengers and vehicles are described according to a prototypical ontology devised for the car pooling case study. An excerpt of the semantic descriptions of a SUV and a Convertible Car follows. All examples are in classical DL notation for the sake of conciseness.

**SUV:** \( \text{Car} \sqcap \exists \text{vehicle\_Type} \sqcap \forall \text{vehicle\_Type.SUV} \sqcap \exists \text{accepts} \sqcap \forall \text{accepts.NonSmoking} \sqcap \exists \text{has\_Feature} \sqcap \forall \text{has\_Feature.}(\text{Car\_Radio} \sqcap \text{Air\_Conditioning} \sqcap \text{Baby\_Seat}) \sqcap \exists \text{comfort\_Level} \sqcap \forall \text{comfort\_Level.High\_Comfort} \sqcap \exists \text{driver\_Experience} \sqcap \forall \text{driver\_Experience.High\_Xp} \sqcap = 6 \text{ available\_Seats} \sqcap = 650 \text{ available\_Capacity} \sqcap \geq 1 \text{ carries\_Luggage} \).

---

7
ConvertibleCar:

\[
\text{Car} \sqcap \exists \text{vehicle.Type} \sqcap \forall \text{vehicle.Type.Convertible.Car} \sqcap \exists \text{has.Feature} \sqcap \forall \text{has.Feature.}(\text{Sun.Roof} \sqcap \neg \text{Baby.Seat}) \sqcap \exists \text{comfort.Level} \sqcap \forall \text{comfort.Level.Standard.Comfort} \sqcap \exists \text{driver.Experience} \sqcap \\
\forall \text{driver.Experience.High.Xp} \sqcap = 1 \text{ available.Seats} \sqcap = 250 \text{ available.Capacity} \sqcap \geq 1 \text{ carries.Luggage}.
\]

In order to take into account passenger-car and passenger-passenger constraints, the framework exploits semantic matchmaking. The system adopts a distributed, greedy first-come first-served approach. The following examples show how it works.

1. **UserA**, headed downtown with her baby, is looking for a car with two seating positions—one of them equipped with baby seat—and enough space for a medium-size luggage and a stroller. She would like to travel comfortably in an air-conditioned, quiet and non-smoking environment. In formulas:

\[
\text{UserA}: \exists \text{vehicle.Type} \sqcap \forall \text{vehicle.Type.SUV} \sqcap \exists \text{accepts} \sqcap \forall \text{accepts.}(\text{NonSmoking} \sqcap \text{Quiet}) \sqcap \\
\exists \text{has.Feature} \sqcap \forall \text{has.Feature.}(\text{Baby.Seat} \sqcap \text{Air Conditioning}) \sqcap \exists \text{comfort.Level} \sqcap \\
\forall \text{comfort.Level.High.Comfort} \sqcap \exists \text{driver.Experience} \sqcap \forall \text{driver.Experience.High.Xp} \sqcap \geq 2 \text{ available.Seats}.
\]

<table>
<thead>
<tr>
<th></th>
<th>Sedan</th>
<th>Convertible</th>
<th>Big City-car</th>
<th>Small Citycar</th>
<th>SUV</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="UserA" /></td>
<td>56.1</td>
<td>31.5</td>
<td>64.3</td>
<td>76.2</td>
<td><strong>90.2</strong></td>
</tr>
</tbody>
</table>

Table 2: Reasoner output for example stage 1

Matching finds which vehicles of the fleet best meet these requirements. Outcomes are summarized in Table 2. **UserA** and **SUV** are close in terms of car direction, because the angle difference between target locations is about 10° (with a supposed threshold of 60°). Furthermore, **UserA’s** semantic description is very similar to **SUV**, because all atomic concepts, universal quantifiers and unqualified number restrictions on roles are compatible. On the other hand, **UserA** is not compatible with the convertible car. An explanation for this is given by the Concept Contraction output:

**Give up:** \forall \text{vehicle.Feature.Baby.Seat} \sqcap \geq 2 \text{ available.seats}

The results are ranked using the utility function in Section 3.2. The best match for **UserA** is with **SUV**, so she is assigned to it.

2. When the passenger/car association is confirmed, the new passenger’s information is appended in conjunction with the **SUV** annotation

3. **UserB** is matched against the updated fleet. *He is looking for a ride to the hills. It is a trip for pleasure that he would like to do with friendly people and maybe music.* He carries only a small luggage, but he is a smoker. Results denote that **UserB** is not compatible with the **luxury sedan** and the **big citycar**, due to his smoking habit. Moreover, in matching with the **SUV**, his requests go against **UserA**’s requirement of silence. Full data are not shown due to limited space, only semantic matchmaking results are reported in Table 4: **UserB** is assigned to the convertible car, whose record is updated accordingly. If needed, the system can show inconsistencies by means of the Concept Contraction inference service offered by the reasoning engine [17]. Outcome explanation is a very important feature and a unique advantage of approaches based on knowledge representation.

<table>
<thead>
<tr>
<th>s.penalty</th>
<th>Sedan</th>
<th>Convertible</th>
<th>Big City-car</th>
<th>Small Citycar</th>
<th><strong>SUV</strong> + <strong>UserA</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="UserB" /></td>
<td>54.2</td>
<td><strong>86.8</strong></td>
<td>49.9</td>
<td>65.7</td>
<td><strong>38.4</strong></td>
</tr>
</tbody>
</table>

Table 3: Reasoner output for example stage 3

4. The same process is repeated with remaining passengers to build other groups. Table 4 shows the results of the computation.

At the end of the matchmaking process the passengers will be arranged on the cars as follows:
<table>
<thead>
<tr>
<th>$s_{\text{penalty}}$</th>
<th>Luxury Sedan + UserD</th>
<th>Convertible + UserB</th>
<th>Big Citycar + UserC + UserE</th>
<th>Small Citycar + UserF</th>
<th>SUV + UserA</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>72.1</td>
<td>64.3</td>
<td><strong>91.2</strong></td>
<td>85.9</td>
<td>60.1</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>85.5</td>
<td>50.1</td>
<td>63.1</td>
<td>45.5</td>
<td>70.9</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>67.5</td>
<td>58.9</td>
<td><strong>86.4</strong></td>
<td>70.1</td>
<td>48.2</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>74.5</td>
<td>11.3</td>
<td>56.0</td>
<td><strong>81.9</strong></td>
<td>80.1</td>
</tr>
</tbody>
</table>

Table 4: Reasoner output for the next 4 users

- SUV: \{UserA\};
- Small citycar: \{UserF\};
- Big citycar: \{UserC, UserE\};
- Luxury sedan: \{UserD\};
- Convertible: \{UserB\};

In any case, explanation for the outcomes can be obtained exploiting the Concept Contraction and Abduction inference services. Relevant examples include:
- For UserC, **Big Citycar** was preferred over **Small Citycar**: they both provided most of the specified features, but the luggage carried by UserC best fitted in **Big Citycar** trunk available/residual capacity.
- The UserF/Big Citycar match had a poor score due to insufficient available seats after accommodating both UserC and UserE.

5 Conclusion and future work

The paper presented a framework for semantic-enhanced discovery of transport resources in car sharing and pooling scenarios. It exploits OpenStreetMap cartography enriched with formal logic-based annotations and an embedded lightweight reasoner to execute semantic matchmaking between user and vehicle profiles. Benefits of the approach have been illustrated in a case study with reference to the feasibility and sustainability of car pooling. Future work includes improvements to the framework with further discovery features and contextual parameters, as well as the development of a full prototype of mobile travel assistant. Field trials will be needed to validate the overall acceptability of the proposed solutions.

References


